



**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN ROBOT PENARI HUMANOID  
DENGAN 25 DOF UNTUK MELAKUKAN GERAKAN TARI  
REMO**

Muchammad Ainur Fahd  
NRP 0711144000041

Dosen Pembimbing  
Dr. Ir. Djoko Purwanto, M.Eng.  
Muhammad Hilman Fatoni, S.T., M.T.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN ROBOT PENARI HUMANOID  
DENGAN 25 DOF UNTUK MELAKUKAN GERAKAN TARI  
REMO**

Muchammad Ainur Fahd  
NRP 07111440000041

Dosen Pembimbing  
Dr. Ir. Djoko Purwanto, M.Eng.  
Muhammad Hilman Fatoni, S.T., M.T.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2018





**FINAL PROJECT - TE 141599**

***DESIGN HUMANOID ROBOT DANCING WITH 25  
DEGREE OF FREEDOM FOR DOING MOVEMENT REMO  
DANCE***

Muchammad Ainur Fahd  
NRP 07111440000041

Supervisor  
Dr. Ir. Djoko Purwanto, M.Eng.  
Muhammad Hilman Fatoni, S.T., M.T.

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Electrical Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2018



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “RANCANG BANGUN ROBOT PENARI HUMANOID DENGAN 25 DOF UNTUK MELAKUKAN GERAKAN TARI REMO” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 17 Juli 2018

Muchammad Ainur Fahd  
NRP. 0711144000041

.....*Halaman ini sengaja dikosongkan*.....



# **RANCANG BANGUN ROBOT PENARI HUMANOID DENGAN 25 DOF UNTUK MELAKUKAN GERAKAN TARI REMO**

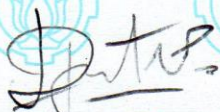
## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada**

**Bidang Studi Elektronika  
Departemen Teknik Elektro  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember**

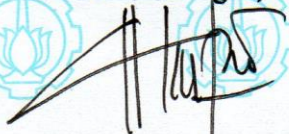
**Menyetujui :**

**Dosen Pembimbing I,**



**Dr. Ir. Djoko Purwanto, M.Eng.**  
**NIP. 196512111990021002**

**Dosen Pembimbing II,**



**Muhammad Hilman Fatoni, S.T., M.T.**  
**NIP. 199103252015041001**



.....*Halaman ini sengaja dikosongkan*.....

# **Rancang Bangun Robot Penari Humanoid dengan 25 DoF untuk Melakukan Gerakan Tari Remo**

Nama : Muchammad Ainur Fahd  
Pembimbing I : Dr. Ir. Djoko Purwanto, M.Eng.  
Pembimbing II : Muhammad Hilman Fatoni, S.T., M.T.

## **ABSTRAK**

Seiring dengan perkembangan zaman, teknologi saat ini berkembang secara pesat. Sebut saja salah satu teknologi tersebut adalah dibidang robot. Saat ini robot mulai banyak diaplikasikan, contohnya seperti untuk mempermudah pekerjaan manusia, hiburan, militer dan masih banyak lagi.

Salah satu jenis robot yang banyak dikembangkan saat ini adalah robot humanoid. Robot humanoid merupakan robot yang hampir menyerupai manusia secara umum dan memiliki pergerakan layaknya manusia pada umumnya seperti berjalan, berdiri dan lain sebagainya. Saat ini banyak negara di dunia yang berlomba-lomba untuk menciptakan robot humanoid yang memiliki kecerdasan buatan layaknya manusia. Salah satunya robot Atlas yang dibuat oleh Boston Dynamics yang bisa melakukan gerakan salto.

Di Indonesia sendiri, perkembangan robot humanoid kebanyakan masih dalam ukuran kecil (*kids size*) dan digunakan untuk perlombaan seperti robot sepak bola dan robot seni tari. Pengembangan robot seni tari sendiri saat ini masih sebatas penyempurnaan gerak, keseimbangan dan komunikasi antar robot. Desain mekanik pada robot terdahulu juga dirasa tidak memenuhi *Center of Gravity* maupun *Center of Mass*. [1] Oleh karena itu perlu dibuat sebuah platform robot tari humanoid yang baru agar lebih luwes untuk melakukan gerakan tarian remo.

Dari hasil pengujian, robot humanoid yang dinamai virose ini lebih kokoh dari segi kontruksi sehingga lebih stabil untuk berjalan maupun menari. Selain itu pergerakan tangan lebih luwes dengan beberapa penambahan frame gerakan. Secara keseluruhan robot dapat melakukan gerakan tari remo yang disesuaikan dengan tarian aslinya.

Kata kunci: Robot humanoid, robot penari, kontes robot seni tari Indonesia.

.....*Halaman ini sengaja dikosongkan*.....

# **DESIGN HUMANOID ROBOT DANCING WITH 25 DEGREE OF FREEDOM FOR DOING MOVEMENT REMO DANCE**

*Name* : Muchammad Ainur Fahd  
*Supervisor I* : Dr. Ir. Djoko Purwanto, M.Eng.  
*Supervisor II* : Muhammad Hilman Fatoni, S.T., M.T.

## **ABSTRACT**

*As time goes by, technology growing rapidly. One of the technology is robotic. Now a days, robots began to be widely applied, for example like to facilitate the work of humans, entertainment, military and much more.*

*One type of robot that is widely developed today is a humanoid robot. Humanoid robot is a robot that almost resembles a human in general and has a movement like a human in general like walking, standing and so forth. Currently many countries in the world are competing to create humanoid robots that have artificial intelligence like humans. One of them is Atlas robot made by Boston Dynamics that can make salto movement.*

*In Indonesia alone, the development of humanoid robots are mostly still in kids size and used for competitions such as soccer robots and dance robots. Development of the robot art of dance itself is still limited to the completion of motion, balance and communication between robots. Mechanical design on the previous robot also felt did not meet the Center of Gravity and Center of Mass.[1] Therefore, it is necessary to create a new humanoid robot dance platform to be more flexible to perform remo dance movements.*

*From the test results, humanoid robot called Virose can be said to be more solid in terms of construction, so it is more stable to walk. In addition, the hand movements are more flexible with some addition of movement frames. Overall robots can perform remo dance moves that are tailored to the original dance.*

*Keywords: Humanoid Robot, Robot dancer, Indonesia dance robot contest.*

.....*Halaman ini sengaja dikosongkan*.....

## KATA PENGANTAR

Alhamdulillah, syukur yang tiada henti penulis panjatkan kehadirat Allah SWT serta tidak lupa sholawat serta salam semoga tetap tercurah kepada junjungan kita Nabi Muhammad SAW sehingga penelitian dalam tugas akhir ini bisa berjalan lancar dan selesai tepat pada waktunya.

Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak, dan penulis sampaikan rasa terima kasih. Terima kasih yang sebesar-besarnya juga penulis sampaikan kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini, diantaranya:

1. Kedua orang tua yang senantiasa memberikan do'a, nasihat, motivasi, dukungan dan karena keberadaan merekalah penulis tetap semangat untuk menyelesaikan penelitian ini.
2. Dr. Ir. Djoko Purwanto, M.Eng., selaku dosen pembimbing, atas dukungan, serta bimbingan, dan pengarahan yang diberikan selama pengerjaan penelitian tugas akhir ini.
3. Muhammad Hilman Fatoni, S.T., M.T., selaku dosen pembimbing II, atas dukungan, serta bimbingan, dan pengarahan yang diberikan selama pengerjaan penelitian tugas akhir ini.
4. Seluruh dosen-dosen Departemen Teknik Elektro, khususnya dosen-dosen bidang studi Elektronika.
5. Seluruh teman-teman penulis terutama teman-teman angkatan e-54 dan laboratorium B202.
6. Seluruh teman-teman beserta dosen pembimbing Tim Robotika ITS khususnya untuk tim VIROSE dan Lab Robot Cerdas.

Penulis sadar bahwa Tugas Akhir ini masih belum sempurna dan masih banyak hal yang perlu diperbaiki. Penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi banyak pihak dan dapat membantu pengembangan tentang robot humanoid kedepannya.

Surabaya, 17 Juli 2017

Muchammad Ainur Fahd

.....*Halaman ini sengaja dikosongkan*.....



## DAFTAR ISI

PERNYATAAN KEASLIAN .....	i
TUGAS AKHIR.....	iii
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL .....	xv
<b>PENDAHULUAN</b> .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah.....	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	3
1.5. Metodologi Penelitian.....	3
1.6. Sistematika Penulisan .....	4
<b>TINJAUAN PUSTAKA</b> .....	7
2.1. Robot Humanoid .....	7
2.2. Kontes Robot Seni Tari Indonesia (KRSTI) .....	7
2.3. Tari Remo.....	8
2.4. Kinematika Robot.....	9
2.5. Platform Robot Humanoid Darwin OP[3] .....	10
2.5.1. Fit-PC2i .....	11
2.5.2. CM-730 .....	12
2.5.3. Servo MX-28T.....	13
2.6. Platform Robot Humanoid Kondo KHR-3HV .....	15
2.6.1. Servo Controller RCB-4HV .....	15
2.6.2. Servo KRS-2552HV .....	16
2.7. Arduino Nano.....	17
2.8. Baterai Lithium Polymer .....	18
2.9. Autodesk Inventor 2018.....	20
<b>PERANCANGAN SISTEM DAN IMPLEMENTASI DESAIN.....</b>	<b>21</b>
3.1. Desain Mekanik Robot .....	21
3.2. Perancangan dan Implementasi Sistem Elektronik .....	23
3.2.1. Power Switching .....	23
3.2.2. Sensor Suara .....	24
3.2.3. Main Controller .....	25

3.2.4. Sub Controller .....	26
3.3. Desain Komunikasi Antar Robot .....	27
3.4. Perancangan Gerakan Tari Remo .....	28
3.4.1 RoboPlus Motion.....	29
3.4.2 HeartToHeart4.....	30
3.5. Implementasi Desain Mekanik Robot.....	30
3.6. Pembuatan Gerakan Tari Remo .....	31
3.7. Implementasi Gerakan Tari Remo .....	34
3.8. Sistem Penilaian KRSTI 2018.....	36
<b>PENGUJIAN DAN ANALISIS.....</b>	<b>37</b>
4.1. Pengujian Sensor Suara.....	37
4.2. Pengujian Komunikasi Main Controller dengan Sub Controller.....	39
4.3. Pengujian Tarian Remo pada Robot.....	40
4.3.1. Pengujian Robot di Lapangan.....	40
4.3.2. Pengujian Robot di KRSTI tingkat Regional.....	42
4.3.3. Pengujian Robot di KRSTI tingkat Nasional.....	43
<b>PENUTUP.....</b>	<b>45</b>
5.1. Kesimpulan.....	45
5.2. Saran.....	45
<b>DAFTAR PUSTAKA.....</b>	<b>47</b>
LAMPIRAN A .....	49
LAMPIRAN B .....	55
LAMPIRAN C .....	64
LAMPIRAN D .....	70
BIODATA PENULIS.....	79

## DAFTAR GAMBAR

Gambar 1.1 Kontes Robot Seni Tari Indonesia. ....	2
Gambar 2.1 Gambaran Pertandingan Kontes Robot Seni Tari Indonesia.8	
Gambar 2.2 Penari Remo[2]. ....	9
Gambar 2.3 Kinematika Robot dengan 3 DoF .....	10
Gambar 2.4 Robot Darwin OP dan Spesifikasi Dimensi .....	11
Gambar 2.5 Framework Darwin-OP .....	11
Gambar 2.6 CM-730 Gyroscope.....	12
Gambar 2.7 Servo MX-28T.....	14
Gambar 2.8 Perbandingan Kecepatan dan Torsi Servo Dynamixel.....	14
Gambar 2.9 Robot Kondo KHR-3HV.....	15
Gambar 2.10 Servo Controller RCB-4HV .....	16
Gambar 2.11 Sistem Komunikasi Servo Serial .....	16
Gambar 2.12 Servo KRS-2552HV .....	17
Gambar 2.13 Arduino Nano V.3 .....	17
Gambar 2.14 Pinout Arduino Nano V.3 .....	18
Gambar 2.15 Battery LiPo Turnigy 2200mAh .....	19
Gambar 2.16 Software Autodesk Inventor 2018 .....	20
Gambar 3.1 Desain Mekanik Robot Humanoid. ....	22
Gambar 3.2 Desain Sistem Elektronik Robot.....	23
Gambar 3.3. Skematik Power Switching .....	24
Gambar 3.4. Realisasi Board Power Switching.....	24
Gambar 3.5. Skematik Sensor Suara.....	25
Gambar 3.6. Realisasi Sensor Suara. ....	25
Gambar 3.7. Main Controller Fit-PC2i .....	26
Gambar 3.8. Board sub controller dengan Arduino Nano.....	27
Gambar 3.9. Komunikasi Antar Robot.....	28
Gambar 3.10. Pola Lantai yang Diterapkan .....	29
Gambar 3.11. Software RoboPlus Motion .....	29
Gambar 3.12. Software HeartToHeart4 .....	30
Gambar 3.13. Desain robot lengkap dengan aksesoris .....	31
Gambar 3.14. Software Roboplus.....	32
Gambar 3.15. Program memanggil motion.....	32
Gambar 3.16. Pose Robot dengan RoboPlus Motion .....	33
Gambar 3.17. Pose Robot dengan HeartToHeart4.....	33
Gambar 4.1. Grafik Output Sensor Suara. ....	37
Gambar 4.2. Grafik Sensor Suara ketika di Mute.....	38

Gambar 4.3. Pembacaan Sensor Suara .....39

Gambar 4.4. Monitoring robot dengan kondisi STOP .....40

Gambar 4.5. Monitoring robot dengan kondisi GO. ....40

## DAFTAR TABEL

Tabel 2.1 Spesifikasi Fit-PC2i.....	12
Tabel 2.2 Spesifikasi CM-730 .....	13
Tabel 2.3 Spesifikasi Arduino Nano V3 .....	18
Tabel 3.1 Spesifikasi Dimensi Robot.....	22
Tabel 3.2 Gerakan Dasar Tari Remo .....	34
Tabel 3.3 Penilaian KRSTI 2018.....	36
Tabel 4.1 Truth Table pada Robot Virose.....	39
Tabel 4.2 Hasil Pengujian di Lapangan Latihan.....	41
Tabel 4.3 Hasil Nilai tingkat Regional.....	42
Tabel 4.4 Hasil Nilai 14 Besar tingkat Nasional .....	43
Tabel 4.5 Hasil Nilai 8 Besar tingkat Nasional .....	43
Tabel 4.6 Hasil Nilai 4 Besar tingkat Nasional .....	44

.....*Halaman ini sengaja dikosongkan*.....

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Teknologi di bidang robotika semakin hari semakin berkembang, hal ini mempengaruhi akan kebutuhan robotika semakin meningkat, seperti bidang industri, sipil, medis, olahraga, manajemen bencana, sampai di bidang kesenian. Riset pada bidang robotika pun terbilang semakin banyak, tentunya hal ini bukan tanpa tujuan. Demikian juga penelitian tentang robot humanoid atau robot yang menyerupai manusia. Robot humanoid ini bisa melakukan banyak tugas ataupun tujuan seperti olahraga, manajemen bencana, dan kesenian.

Di Indonesia sendiri penelitian tentang robot humanoid semakin pesat. Hal ini dikarenakan adanya perlombaan baik itu tingkat Regional, Nasional, bahkan Internasional. Kontes Robot Indonesia (KRI) adalah suatu kegiatan perlombaan yang diselenggarakan oleh Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia (Kemristek dikti). Pada KRI terdapat berbagai divisi, salah satunya adalah Kontes Robot Seni Tari Indonesia (KRSTI). Divisi ini merupakan suatu ajang kompetisi perancangan, pembuatan, dan pemrograman robot yang disertai dengan unsur-unsur seni dan budaya bangsa Indonesia khususnya seni tari yang telah terkenal di bumi pertiwi. Untuk KRSTI 2018, kembali guna membangkitkan kecintaan dan pelestarian budaya-budaya Nasional maka tema yang diangkat adalah Robot Penari Remo.[2]

Pada pertandingan Kontes Robot Seni Tari Indonesia, robot diharuskan dapat melakukan beberapa gerakan tarian ketika ada alunan musik dan berhenti ketika tidak ada alunan musik. Musik diperdengarkan melalui Audio Bluetooth Transmitter dari juri dan didengarkan oleh robot dengan Audio Bluetooth Receiver. Selain itu, robot harus melakukan tarian mulai dari zona Start – zona A – zona B – zona C – zona Finish. Dimana tiap zona harus dilewati untuk mendapatkan point nilai dan bonus.

Dalam rangka pengembangan Robot Seni Tari humanoid, penelitian ini dilakukan dengan menggunakan 25 DoF (*Degree of Freedom*) yang diharapkan dapat melakukan gerakan tari remo. Pengembangan Robot Seni Tari sendiri saat ini masih sebatas penyempurnaan gerak keseimbangan dan komunikasi antar robot.

Desain mekanik terdahulu dirasa tidak mumpuni untuk melakukan gerakan tari remo dan dengan rule saat ini. Dimana robot harus melakukan beberapa variasi gerakan dengan panjang lintasan tiga meter dengan durasi musik hanya 2 menit 54 detik. Oleh karena itu perlu dibuat *platform* robot tari humanoid yang baru agar dapat melangkah lebih jauh dan lebih luwes ketika melakukan gerak tarian remo.



**Gambar 1.1** Kontes Robot Seni Tari Indonesia.

## **1.2. Perumusan Masalah**

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Perancangan robot penari humanoid yang mampu mengimplementasikan gerakan tari remo.
2. Sistem elektronik pada robot penari humanoid.
3. Sistem komunikasi antara dua robot penari humanoid.

## **1.3. Tujuan Penelitian**

Penelitian pada tugas akhir ini bertujuan sebagai berikut:

1. Terciptanya robot penari humanoid yang mampu melakukan gerakan tari remo.
2. Memperoleh sistem elektronik yang dapat menjalankan aktifitas robot penari humanoid.
3. Mengatasi masalah komunikasi antar dua robot penari humanoid



yang selama ini masih belum stabil.

#### **1.4. Batasan Masalah**

Batasan masalah dari tugas akhir ini adalah sebagai berikut:

1. Desain mekanik robot dengan platform yang baru.
2. Robot melakukan gerakan tari remo dengan perintah audio Bluetooth dan berhenti ketika tidak ada musik.
3. Robot hanya melakukan gerakan tari remo.

#### **1.5. Metodologi Penelitian**

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

##### **1. Studi literatur**

Studi literatur berisi serangkaian kegiatan pengumpulan dan pengkajian dasar teori yang relevan dan terpercaya untuk menunjang penulisan tugas akhir ini. Literatur dapat bersumber dari paper, jurnal, artikel, buku, maupun *website*, yang bertaraf nasional dan internasional seperti IEEE Explorer, serta dari hasil konsultasi dengan dosen pembimbing.

##### **2. Pengerjaan Mekanik Robot**

Pada tahap ini robot akan dibuat sebagian besar bagiannya menggunakan alumunium. Bagian bagian robot yang tidak membutuhkan struktur kuat akan dibuat menggunakan 3D Printing. Sebagian besar pekerjaan akan memanfaatkan mesin laser CNC agar didapatkan hasil potongan plat alumunium dan 3D Printing yang presisi dan selesai dalam waktu singkat.

##### **3. Pembuatan Elektronik dan Programming**

Pada tahap ini akan dibuat sebuah system elektronik yang nantinya akan dipasang pada robot. Pemasangan komponen elektrinik meliputi pemasangan sensor, controller, servo, microcontroller, dan komponen-komponen elektronik penunjang lainnya. Pada tahap ini ditargetkan robot dan PC sudah dapat berkomunikasi satu sama lain. Selain itu juga dibuat program supaya fungsi dasar robot seperti berjalan, menari, dan membaca sensor dapat dilakukan.

##### **4. Pengujian dan Analisa**

Pada tahap pengujian meliputi seluruh aktifasi gerakan robot akan diuji apakah sudah bekerja dengan baik dan semestinya.

Pengujian akan dilakukan secara bertahap. Pada tahap awal robot diuji coba untuk gerakan berjalan dilapangan terlebih dahulu. Selanjutnya mulai melakukan pengujian terhadap system komunikasi antara dua robot. Pengujian pada lapangan akan dilakukan sesuai dengan keadaan pertandingan yang sebenarnya. Hasil pengembangan robot juga akan dibandingkan dalam Kontes Robot Seni Tari Indonesia tingkat Regional dan Nasional.

## **5. Analisa dan Evaluasi**

Analisa dilakukan terhadap hasil pengujian sehingga didapatkan hasil yang seharusnya. Evaluasi dilakukan untuk memperbaiki system yang sudah dibuat. Bahasan utama dalam evaluasi antara lain tentang desain mekanik pada robot apakah sudah bisa melakukan gerakan tari remo dan system komunikasi antara dua robot sudah stabil. Dengan evaluasi seperti itu, system akan selalu dioptimasi sehingga dapat memenuhi target yang diinginkan.

## **6. Penyusunan Laporan Tugas Akhir**

Tahap penyusunan laporan merupakan tahap terakhir dari proses pengerjaan tugas akhir ini. Laporan berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, studi literatur, tinjauan pustaka, perancangan dan pembuatan sistem, pengujian dan analisa, serta penutup.

### **1.6. Sistematika Penulisan**

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi Lima Bab dengan sistematika penulisan sebagai berikut:

- **Bab I : Pendahuluan**  
Bab ini meliputi latar belakang, perumusan masalah, tujuan penelitian, Batasan masalah, metodologi, sistematika penulisan, dan relevansi.
- **Bab II : Tinjauan Pustaka**  
Bab ini menjelaskan tentang teori penunjang yang terkait maupun yang dibutuhkan dalam pengerjaan tugas akhir ini.
- **Bab III : Perancangan Sistem dan Implementasi**  
Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah sehingga menghasilkan visualisasi. Guna mendukung itu digunakanlah blok diagram agar mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.

- Bab IV : Pengujian dan Analisis  
Bab ini menjelaskan tentang hasil uji coba alat beserta analisisnya. Selain itu juga akan ditampilkan beberapa visualisasi pada robot dari hasil pengamatan.
- Bab V : Penutup  
Bab ini berisi tentang kesimpulan yang diperoleh dari pembuatan alat serta saran untuk pengembangan lebih lanjut.

.....*Halaman ini sengaja dikosongkan*.....

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Robot Humanoid**

Robot humanoid adalah robot yang berpenampilan keseluruhan berdasarkan tubuh manusia, dapat melakukan interaksi terhadap peralatan maupun lingkungan disekitar kita. Secara umum robot humanoid terdiri dari kepala, badan, dua lengan, dan dua kaki. Posisi-posisi sendi gerak pada robot humanoid mengadaptasi bagian-bagian sendi manusia. Jika manusia memiliki sendi, dirobot menggunakan servo.

Saat ini penggunaan robot humanoid sudah sangat berkembang di dunia. Penggunaannya sendiri mulai dari olahraga, manajemen keselamatan, militer, hingga kesenian. Dalam pengendalian dan perencanaan antara robot jenis humanoid dengan robot yang ada di industri memiliki beberapa perbedaan yaitu pada gerakan robot yang hampir mirip dengan manusia, menggunakan kaki sebagai alat berjalan, dapat melakukan tugas seperti manusia. Sedangkan pada bidang industri hanya melakukan tugas yang sudah direncanakan, proses berjalannya pun dibantu dengan roda. Untuk alasan tersebut, studi tentang kontrol dan dinamika jadi lebih penting. Untuk menjaga keseimbangan pada robot humanoid perlu beberapa cara, seperti melihat Center of Gravity, Center of Mass, dan Zero Moment Point (ZMP).

Beberapa robot humanoid memiliki sensor yang dapat mengumpulkan data informasi pada dunia nyata dan dapat berinteraksi dengan itu. Ada fitur pada manusia yang belum ditemukan atau masih dalam pengembangan seperti tentang keselamatan robot itu sendiri atau orang-orang. Pada penggunaanya, robot humanoid masih dalam tahap penyempurnaan.

#### **2.2. Kontes Robot Seni Tari Indonesia (KRSTI)**

Kontes Robot Indonesia (KRI) adalah suatu kegiatan perlombaan yang diselenggarakan oleh Kementerian Riset, Teknologi, dan Pendidikan Tinggi Republik Indonesia (Kemendiknas). Pada KRI terdapat berbagai divisi, salah satunya adalah Kontes Robot Seni Tari Indonesia (KRSTI). Divisi ini merupakan suatu ajang kompetisi perancangan, pembuatan, dan pemrograman

robot yang disertai dengan unsur-unsur seni dan budaya bangsa Indonesia khususnya seni tari yang telah terkenal di bumi pertiwi. Untuk KRSTI 2018, kembali guna membangkitkan kecintaan dan pelestarian budaya-budaya Nasional maka tema yang diangkat adalah Robot Penari Remo.

Pada pertandingan Kontes Robot Seni Tari Indonesia, robot diharuskan dapat melakukan beberapa gerakan tarian ketika ada alunan musik dan berhenti ketika tidak ada alunan musik. Musik diperdengarkan melalui Audio Bluetooth Transmitter dari juri dan didengarkan oleh robot dengan Audio Bluetooth Receiver. Selain itu, robot harus melakukan tarian mulai dari zona Start – zona A – zona B – zona C – zona Finish. Dimana tiap zona harus dilewati untuk mendapatkan point nilai dan bonus.



**Gambar 2.1** Gambaran Pertandingan Kontes Robot Seni Tari Indonesia.

### **2.3. Tari Remo**

Tari remo merupakan jenis tari putra tunggal yang berasal dari Jawa Timur. Remo berasal dari tari putra namun berkembang juga menjadi putri atau dilembutkan, tapi karakter tegas tetap diterapkan. Masyarakat sudah banyak mengenal tari remo ini, menurut proses kelahiran tari ini diawali dari Ludruk atau pentas seni humoris yang berasal dari Surabaya. Tari remo biasa digunakan sebagai tarian pembuka pada suatu acara. Tari remo sendiri menggambarkan sikap

seorang satria yang gagah perkasa, jujur, dan berani. Hal itu terlihat dari wujud gerak yang tegas dan berkarakter pada penari. Karakteristik yang paling utama dari tata gerak Tari Remo adalah gerakan kaki yang rancak dan dinamis. Gerakan tersebut ditambah dengan adanya bandul lonceng saat penari memainkan kakinya atau saat berjalan. Selain itu, ciri khas yang lain adalah gerakan melempar selendang atau sampur secara cepat dan dinamis, gerakan anggukan dan gelengan kepala, ekspresi wajah, serta kuda-kuda penari membuat tarian ini semakin atraktif.



**Gambar 2.2** Penari Remo[2].

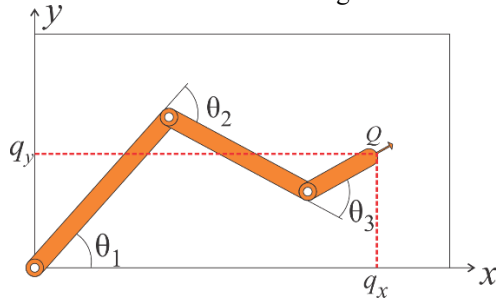
Tata busana tari remo sendiri bermacam-macam menurut wilayah kebudayaannya. Gaya-gaya busananya seperti gaya surabayan, malangan, jombangan, sawunggaling dan remo putri. Irama musik yang mengiringi Tari Remo ini adalah seperangkat gamelan yang berpadu dengan beberapa kethuk, kenong, dan gong.

## **2.4. Kinematika Robot**

Kinematika adalah analisa tentang gerak yang mengacu pada referensi koordinat tanpa memperhatikan faktor-faktor gaya yang menyebabkan pergerakan tersebut. Model kinematika mempresentasikan hubungan antara end effector dalam ruang tiga dimensi. Dalam kinematika robot, terdapat dua macam yaitu *forward kinematics* dan *inverse kinematics*. Gambar 2.3 adalah contoh kinematika pada robot dengan 3 DoF.

Pada Forward Kinematic data yang menjadi inputan adalah sudut pada masing-masing *joint* sedangkan data outputan berupa posisi

koordinat. Sehingga posisi masing-masing joint dan model struktur mekanik robot. Sedangkan Inverse Kinematics, data yang menjadi inputan merupakan posisi koordinat kartesian dan data outputnya berupa sudut pada masing-masing joint. Sehingga Inverse Kinematics dan Forward Kinematics saling berkebalikan.

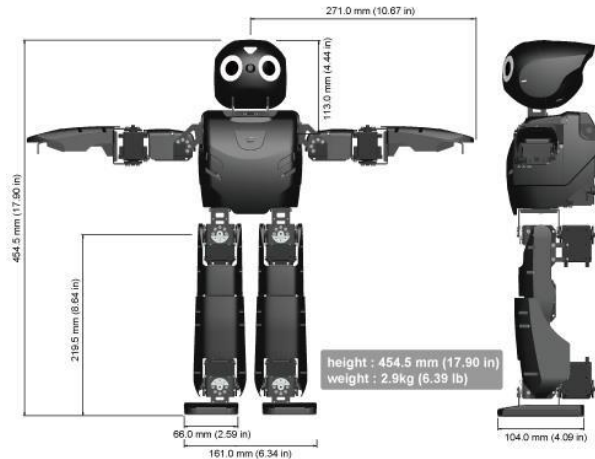


**Gambar 2.3** Kinematika Robot dengan 3 DoF

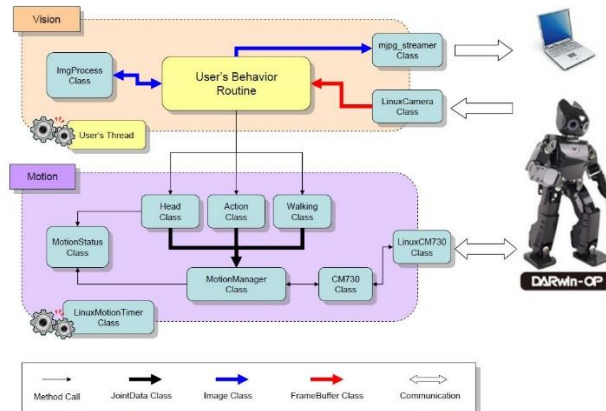
## 2.5. Platform Robot Humanoid Darwin OP[3]

Darwin-OP merupakan kepanjangan dari Dynamic Anthropomorphic Robot with Intelligence Open Platform, merupakan robot yang diproduksi oleh Robotis atas kerjasama dengan University Pennsylvania. Robot ini sengaja dikembangkan untuk keperluan riset dalam permainan speak bola humanoid. Robot Darwin-OP memiliki perangkat keras yang terdiri dari sebuah PC yang bernama Fit-PC2i, kamera Logitech C905, motor servo MX-28T dan sub controller CM-730. PC pada robot sendiri menggunakan Operation System berbasis Linux. PC dapat disambungkan melalui jaringan nirkabel Wi-Fi. Modul sub controller CM-730 merupakan mikrokontroller berbasis ARM yang digunakan untuk mengontrol seluruh servo MX-28T pada robot Darwin. CM-730 memiliki gyroscope dan accelerometer yang berfungsi untuk menstabilkan gerakan robot dan mendeteksi robot jatuh. Darwin OP ini merupakan open platform, dimana seluruh informasi terkait sistem, mekanik, dan sebagainya sudah di publikasikan. Jadi dimungkinkan untuk membuat sebuah robot Darwin OP yang lebih murah.





**Gambar 2.4** Robot Darwin OP dan Spesifikasi Dimensi/4/



**Gambar 2.5** Framework Darwin-OP/4/

### 2.5.1. Fit-PC2i

Fit-PC2i merupakan sebuah mini PC (personal computer) yang menggunakan Intel Atom Z510 32 bit single core dengan clock speed 1.1GHz. Berikut adalah spesifikasi dari Fit-PC2i

**Tabel 2.1.** Spesifikasi Fit-PC2i

Processor	
Type	Intel Atom Z530
Core	32-bit single core
Clock Speed	1.1 GHz
Memory	1 GB DDR2-533
Storage	1x SATA 4GB 2.5" SSD
Graphics	Intel Graphics Media Accelerator GMA 500
Networking	2x GbE LAN ports
Connectivity	2x USB 2.0 (type A USB port) 2x USB 2.0 (mini USB port)
Serial	RS232 via mini serial connector
Operating System	Linux Mint 32-bit
Input Voltage	10 -15 VDC 1.5A

### 2.5.2. CM-730

CM-730 merupakan sub controller yang digunakan oleh robot Darwin OP yang mempunyai fungsi sebagai pengontrol seluruh servo MX-28T. CM-730 ini pada dasarnya berbasis ARM Cortex 32-bit CPU (clocked @72MHz).



**Gambar 2.6** CM-730

Untuk mengakses CM-730 perlu sebuah program yang ditempatkan pada main controller Fit-PC2i. Berikut adalah beberapa fitur dari Sub controller CM-730

**Tabel 2.2.** Spesifikasi CM-730

Fitur	Spesifikasi
CPU	STMicroelectronics 32F103RE ARM Cortex 32-bit CPU (clocked @ 72 MHz) (512KB Flash, 64KB SRAM)
Interface	5x LED, 2x RGB, 3x Button, 1x Buzzer
External ports	13x ADC / I/O Ports
Sensor	3-axis gyroscope, 3-axis accelerometer, supply voltage sensor
Communication	USB Port, Serial Port, 5x TTL Ports (for dynamixel)
Audio & Mic Amp	Audio amp gain: 20, mic amp gan 2027
Additional	Dynamixel Power Control Unit, Head board port, Interface board port
Supply Voltage	8V – 16.8 V
Current Consumption	150mA
Dimension	80.0 mm x 75.0 mm x 20.0 mm
Operation temperature	-65°C to + 80°C
Weight	51g

CM-730 ini memiliki 10 Bit ADC dengan range 0 – 1023, untuk port ADC perlu dimodifikasi dengan penambahan female header.

### **2.5.3. Servo MX-28T**

Servo MX-28T merupakan servo keluaran Robotis, salah satu perusahaan kit robot dan perangkat pendukungnya dari Korea Selatan. Servo ini bisa digolongkan servo cerdas karena pada masing-masing motor terdapat mikrokontroller yang menyediakan kemampuan komunikasi dengan sub controller, dapat melihat posisi, dan beban yang bekerja serta temperatur pada motor.

Dengan banyaknya fitur ini, maka wajar saja jika servo ini tergolong servo mahal. Sebenarnya servo ini memiliki beberapa seri, seperti servo AX, RX, dan yang terbaru MX. Komunikasi antar servo menggunakan komunikasi half duplex TTL, sedangkan kontroller pada tiap servo menggunakan Cortex M3 dengan encoder magnetic

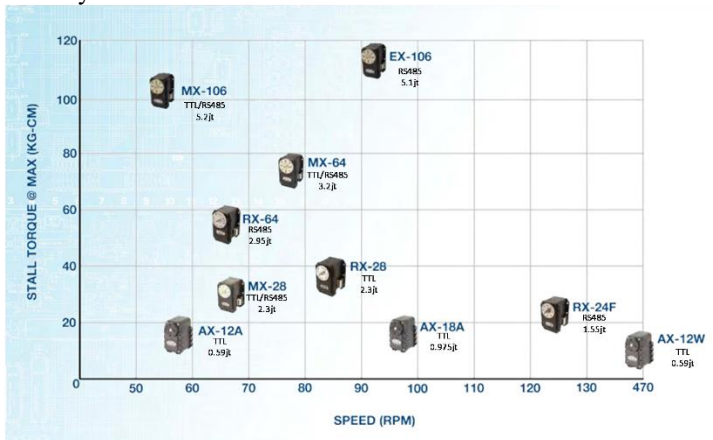
untuk mengetahui posisi servo.

Parameter torsi dan kecepatan pada servo ini bisa terbilang sangat mumpuni. Kecepatan dapat menentukan seberapa cepat putaran servo sedangkan torsi adalah kekuatan servo untuk memutar beban pada porosnya.



**Gambar 2.7** Servo MX-28T

Berikut adalah parameter kecepatan dan torsi untuk setiap servo dari Dynamixel ini :

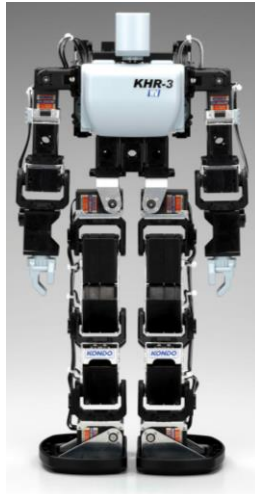


**Gambar 2.8** Perbandingan Kecepatan dan Torsi Servo Dynamixel/4/

## 2.6. Platform Robot Humanoid Kondo KHR-3HV

Robot humanoid Kondo KHR-3HV merupakan robot humanoid yang diproduksi oleh perusahaan Kondo Kagaku co, perusahaan ini berasal dari Jepang. Robot humanoid ini memiliki desain awal sebanyak 17 buah Degree of Freedom (DoF) berupa servo serial. Kemudian desain robot ini dikembangkan agar mampu melakukan gerakan yang lebih beragam menjadi 25 sendi gerak dengan berupa 21 servo serial dan 4 buah servo analog dengan kendali menggunakan *Pulse Modulation Width* (PWM).

Secara umum di Jepang, robot humanoid jenis KHR-3HV digunakan untuk robot *battle* (Robot Petarung) dengan kendali secara manual menggunakan *remote control*.



**Gambar 2.9** Robot Kondo KHR-3HV/5/

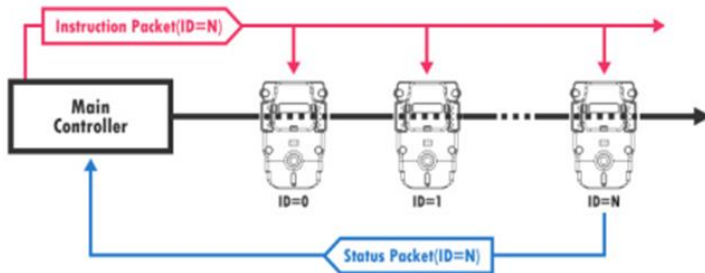
### 2.6.1. Servo Controller RCB-4HV

RCB-4HV merupakan sebuah kontroller untuk menggerakkan servo jenis KRS-2552HV. Untuk servo KRS-2552HV sistem pengaturannya menggunakan sistem komunikasi serial. Kontroller yang digunakan untuk mengatur gerakan servonya adalah RCB-4HV. Controller ini memiliki 1 output serial servo, 8 port serial input output servo, 10 input analog dan 10 port GPIO.



**Gambar 2.10** Servo Controller RCB-4HV

Setiap servo memiliki sistem pengaturannya sendiri-sendiri. Ada dua jenis pengaturan servo, yaitu secara komunikasi serial dan secara analog. Sistem dengan komunikasi serial didesain dengan mengirimkan memberikan alamat pada tiap motor servo dan untuk pengaturannya dengan mengirimkan alamat servo dan posisi servonya. Sedangkan servo dengan pengaturan analog menggunakan *pulse modulation width* (pwm).



**Gambar 2.11** Sistem Komunikasi Servo Serial

### 2.6.2. Servo KRS-2552HV

Sistem pergerakan dan persendian robot KHR-3HV menggunakan motor servo. Motor servo merupakan aktuator gerak secara rotasi yang dirancang dengan pengaturan *close loop* (umpan balik), sehingga dapat ditentukan atau dikendalikan posisi sudut dari porosnya. Bagian dalam dari motor servo terdiri dari beberapa gear,

rangkain dan potensiometer. Rangkaian dan potensiometer digunakan untuk mengatur umpan balik kendalanya dan juga digunakan untuk menentukan batas posisi putaran servonya. Gear pada motor servo digunakan untuk mengatur torsi dari servonya itu sendiri.

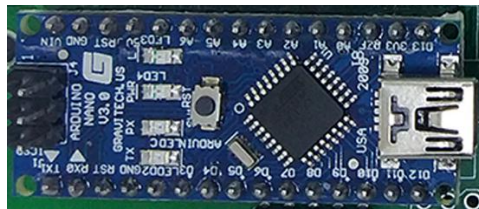
Jenis servo yang digunakan pada robot KHR-3HV adalah servo KRS-2552HV. Servo ini dikontrol dengan sistem komunikasi serial. Sistem ini memudahkan sistem pengkabelan karena dalam sistem serial antar servo dapat dijadikan penghubung.



**Gambar 2.12** Servo KRS-2552HV

## **2.7. Arduino Nano**

Arduino nano merupakan salah satu mikrokontroller yang berbasis ATmega328 (untuk versi 3.x) dan ATmega168 (untuk versi 2.x). Arduino nano memiliki dimensi yang paling kecil daripada jenis Arduino lainnya. Arduino nano dapat diaktifkan melalui koneksi USB mini B atau melalui sumber tegangan eksternal yang belum teregulasi sekitar 6-20 Volt. Arduino nano memiliki chip FTDI FT232L untuk berkomunikasi serial dengan PC.



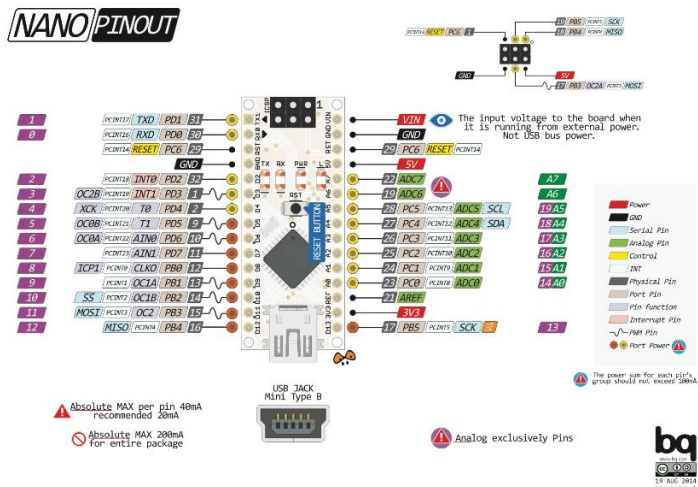
**Gambar 2.13** Arduino Nano V.3

Spesifikasi Arduino dapat dilihat pada tabel 2.3 dan detail pinout

Arduino Nano ini dapat dilihat pada gambar 2.14.

**Tabel 2.3.** Spesifikasi Arduino Nano V.3

Mikrokontroler	Atmel ATmega168 / ATmega328
Tegangan Operasi	5V
Input Voltage (disarankan)	7-12V
Maksimal Voltage	6-20V
Pin Digital I/O	14
Pin Input Analog	8
Arus DC per pin I/O	40 mA
Flash Memory	32KB
SRAM	2KB
EEPROM	1KB
Clock Speed	16 MHz
Dimensi	1.85cm x 4.3cm



**Gambar 2.14** Pinout Arduino Nano V.3 [6]

## 2.8. Baterai Lithium Polymer

Baterai Lithium Polimer atau biasa disebut dngan LiPo merupakan salah satu jenis baterai yang sering digunakan dalam



dunia RC. Utamanya untuk RC tipe pesawat dan helikopter. Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Gambar 2.15 menunjukkan baterai lipo yang dijual di pasaran dengan merk Turnigy yang berkapasitas 2200mAh 3S. Ada tiga kelebihan utama yang ditawarkan oleh baterai berjenis LiPo daripada baterai jenis lain seperti NiCad atau NiMH yaitu:

1. Baterai LiPo memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran.
2. Baterai LiPo memiliki kapasitas penyimpanan energi listrik yang besar.
3. Baterai LiPo memiliki tingkat discharge rate energi yang tinggi, dimana hal ini sangat berguna sekali dalam bidang RC.



**Gambar 2.15** Battery LiPo Turnigy 2200mAh

Selain memiliki keuntungan, baterai ini juga memiliki beberapa kelemahan seperti:

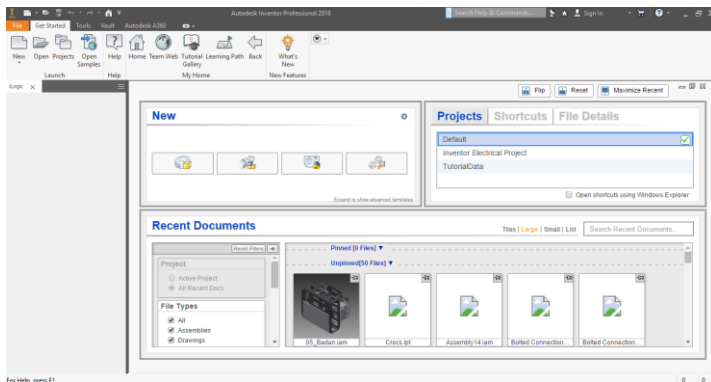
1. Harga baterai LiPo masih tergolong mahal jika dibandingkan dengan baterai jenis NiCad dan NiMH
2. Performa yang tinggi dari baterai LiPo harus dibayar dengan umur yang lebih pendek. Usia baterai LiPo sekitar 300-400 kali siklus 31 pengisian ulang. Sesuai dengan perlakuan yang diberikan pada baterai.
3. Alasan keamanan. Baterai LiPo menggunakan bahan elektrolit yang mudah terbakar.
4. Baterai LiPo membutuhkan penanganan khusus agar dapat bertahan lama. Charging, Discharging, maupun

penyimpanan dapat mempengaruhi usia dari baterai jenis ini.

## 2.9. Autodesk Inventor 2018

Autodesk Inventor 2018 merupakan sebuah perangkat lunak (*software*) yang digunakan untuk membuat desain 3D. File ekstension dari software ini adalah CAD (Computer Aided Drawing). Inventor sendiri adalah produk dari Autodesk Inc. USA atau biasa dikenal dengan AutoCAD. Software ini menyediakan beberapa fitur lengkap yang menunjang permodelan 3D, perakitan (assembly), gambar kerja (Drawing), dan juga animasi yang dapat mempresentasikan desain kita.

Penggunaan Autodesk Inventor ini sangat berguna untuk membuat desain 3D sebelum masuk ke fabrikasi. Detail tiap komponen bisa dibuat atau dimodifikasi dengan software ini. Berikut adalah penampakan dari Autodesk Inventor 2018.



**Gambar 2.16** Software Autodesk Inventor 2018

## **BAB III**

### **PERANCANGAN SISTEM DAN IMPLEMENTASI DESAIN**

Dalam bab ini akan dijelaskan tentang desain robot, perancangan system dan implementasi secara keseluruhan, mulai dari desain mekanik robot, desain sistem elektronik, desain komunikasi pada robot, perancangan gerakan tari remo, implementasi desain mekanik robot, implementasi sistem elektronik, dan implementasi komunikasi pada robot. Perangkat keras yang digunakan adalah dua buah robot humanoid Virose Master dan Virose Slave yang dirancang dengan 25 derajat kebebasan dengan menggunakan beberapa kontroller seperti Fit-PC2i sebagai komputer utama dengan sistem operasi Linux dan mikrokontroller CM-730 sebagai kontroller aktuator servo dynamixel MX-28T serta mikrokontroller Kondo RCB-4HV sebagai kontroller aktuator servo Kondo. Karena pada dasarnya robot humanoid ini didesain menggunakan 2 platform yang berbeda. Ditambah lagi mikrokontroller Arduino Nano sebagai pengolah data dari sensor suara. Dalam sistem ini, komputer utama mengolah data motion dan mengirimkannya ke mikrokontroller CM-730 untuk mengirimkan perintah sudut ke servo dynamixel MX-28T. Sedangkan mikrokontroller Kondo RCB-4HV mengolah data motion dan mengirimkan langsung ke servo Kondo. Pada perancangan robot humanoid ini mengacu pada rule Kontes Robot Seni Tari Indonesia 2018.

#### **3.1. Desain Mekanik Robot**

Desain mekanik robot humanoid ini bertujuan untuk mengikuti pertandingan Kontes Robot Seni Tari Indonesia 2018 dengan mengikuti beberapa ketentuan dari para dewan juri. Pada dasarnya robot humanoid yang akan dirancang menggunakan dua *platform* yang berbeda yaitu dengan platform Darwin OP dari Robotis dan KHR-3HV dari Kondo Robot. Platform Darwin OP digunakan karena kelebihanannya pada kekuatan dan kecepatan servo yang digunakan, sehingga cocok untuk dijadikan tumpuan kaki dan daerah dada. Sedangkan platform Kondo digunakan karena kelebihanannya pada sisi berat yang ringan dan dimensi yang tidak terlalu besar dibandingkan servo dari Darwin OP.

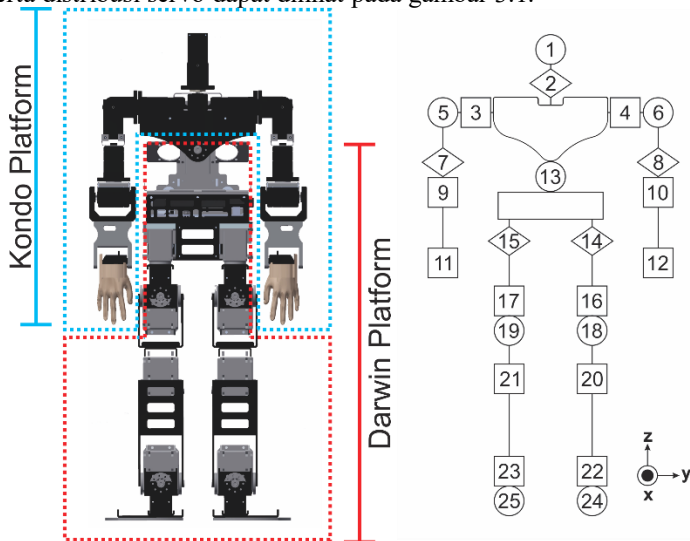
Untuk desain mekanik ini sendiri merujuk pada buku panduan

Kontes Robot Seni Tari Indonesia yang meliputi beberapa ketentuan seperti berikut :

**Tabel 3.1.** Spesifikasi dimensi robot.

No	Deskripsi	Status	
		Ukuran Maksimal	Ukuran Robot
1	Tinggi robot	55±5 cm	54 cm
2	Rentang tangan dari ujung-ujung jari	60 cm	54 cm
3	Lebar telapak kaki	150 cm <sup>2</sup>	11.5 cm <sup>2</sup>
4	Berat robot	30 Kg	3 Kg
5	Derajat Kebebasan	23 DoF (minimal)	25 DoF

Dari tabel diatas, robot humanoid ini bisa dikatakan sesuai dengan peraturan yang ada. Desain robot, diagram derajat kebebasan serta distribusi servo dapat dilihat pada gambar 3.1.



**Gambar 3.1** Desain Mekanik Robot Humanoid.

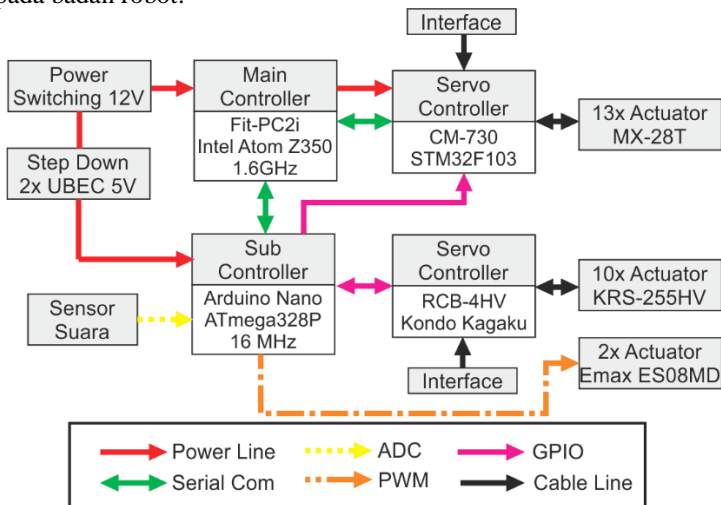
Pada gambar 3.1 derajat kebebasan berjumlah 25 aktuator dengan distribusi nomor 1 sampai 12 menggunakan servo KRS-2552HV dari platform Kondo sedangkan nomor 13 sampai 25

menggunakan servo Dynamixel MX-28T dari platform Robotis.

Frame pada robot humanoid ini menggunakan aluminium tipe 1100 dan 6600. Aluminium tipe 1100 memiliki tingkat kekuatan 99.69 MPa sedangkan tipe 6600 memiliki tingkat kekuatan sebesar 393.48 MPa [7] sehingga cocok untuk diaplikasikan pada frame robot humanoid yang menopang berat beberapa servo dan tidak masalah apabila robot jatuh.

### 3.2. Perancangan dan Implementasi Sistem Elektronik

Robot yang telah dirancang tidak bisa berjalan sesuai yang kita harapkan apabila tidak ada kecerdasan buatan yang ditanam pada robot. Oleh karena itu, robot memerlukan sistem elektronik yang dapat menjalankan seluruh program. Diagram sistem elektronik dapat dilihat pada gambar 3.2. Seluruh komponen elektronik ini diletakkan pada badan robot.

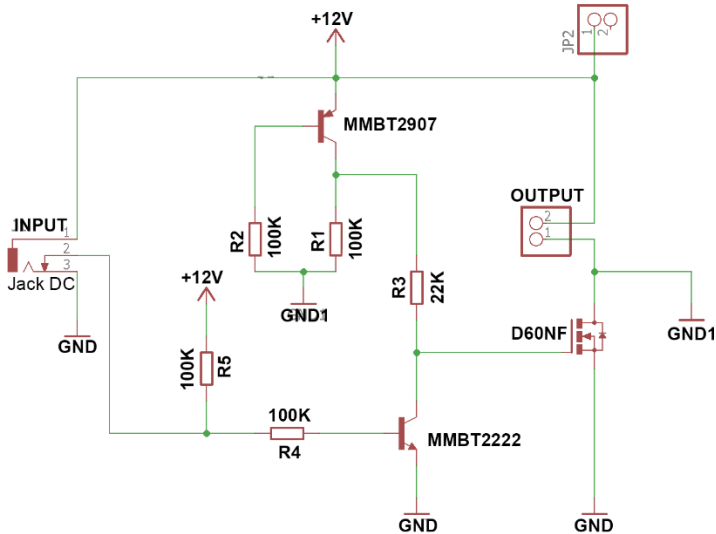


**Gambar 3.2** Desain Sistem Elektronik Robot

#### 3.2.1. Power Switching

Power Switching digunakan sebagai media sumber tegangan yang berasal dari baterai Lithium Polymer atau adaptor. Robot ini menggunakan baterai Li-Po 3 cell atau 12,6v dengan kapasitas 2200 mAh. Saat robot akan menari dilapangan, robot akan menggunakan

baterai Lithium Polymer, sehingga power switch otomatis akan memilih baterai tersebut. Sedangkan saat robot istirahat akan menggunakan adaptor 12v. Skematik dan realisasi board dapat dilihat pada gambar 3.3. Untuk sumber tegangan 5v menggunakan UBEC (Universal Battery Elimination Circuit) yang akan mensupply beberapa board seperti Arduino Nano, dan Sensor Suara. Sedangkan UBEC yang satu lagi untuk mensupply 2 Servo Emax ES08MA II.



**Gambar 3.3. Skematik Power Switching**



**Gambar 3.4. Realisasi Board Power Switching**

### 3.2.2. Sensor Suara

Pada Kontes Robot Seni Tari Indonesia, robot diharuskan dapat

The diagram shows a 5V LED indicator circuit with an audio input. The circuit is powered by a +5V supply and grounded (GND). The input is labeled 'INPUT Audio'. The circuit components and their values are as follows:

- Resistors:** R1 (4K7), R2 (100), R3 (5K), R4 (50K), R5 (100K), R6 (220K), R7 (4K7), R8 (10K), R9 (330), R10 (330).
- Capacitors:** C1 (1nF), C2 (470nF), C3 (470nF), C4 (1nF).
- Transistors:** Q1 (BC548), Q2 (BC548).
- Diodes:** D1 (1N4148).
- LEDs:** Led Indicator, Led Status.

The circuit is designed to turn on the LED indicator when the audio input is present. The output is labeled 'OUTPUT' and 'Led Indicator'.

### 3.2.3. Main Controller

25

berfungsi sebagai pemroses utama dengan sistem operasi berbasis linux. PC ini juga berfungsi untuk membuat motion dengan software RoboPlus Motion yang selanjutnya akan diproses oleh subcontroller CM-730 yang berbasis ARM untuk mengontrol seluruh motor servo MX-28T sebanyak 13 buah. Main Controller ini menerima data serial dari Arduino Nano dan CM-730. Data serial tersebut selanjutnya akan diolah di Main Controller untuk dijadikan referensi untuk menjalankan Servo atau menghentikannya ketika tidak ada lagu.

Main Controller ini menggunakan processor Intel Atom Z530 dengan Clock Speed 1.6GHz, 1GB RAM, 3.6GB Storage Solid State NAND, 2 eksternal USB Ports, HDMI video output, dan networking wifi 802.11b/g/n. Main controller dapat dilihat pada gambar 3.7.



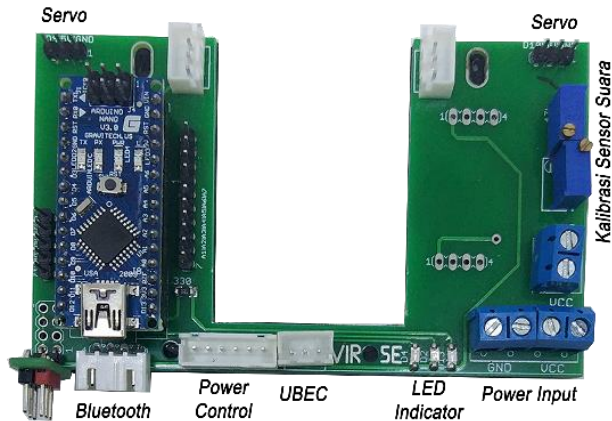
**Gambar 3.7.** Main Controller Fit-PC2i

#### **3.2.4. Sub Controller**

Sub Controller pada robot humanoid virose ini menggunakan dua board, yaitu Arduino Nano dan CM-730. Sub controller ini mempunyai fungsi yang berbeda, Arduino Nano berfungsi sebagai pengolah data sensor suara sedangkan CM-730 berfungsi untuk mengontrol servo Dynamixel MX-28T pada masing-masing joint robot sehingga CM-730 ini bisa disebut servo controller juga. Selain untuk mengontrol servo MX-28T, CM-730 juga mengirimkan data



tombol dari interface untuk dikirimkan ke main controller. Kedua sub controller ini terhubung ke main controller menggunakan jalur komunikasi serial dan dapat dimonitor melalui main controller apabila ada error saat proses pengiriman.



**Gambar 3.8.** Board sub controller dengan Arduino Nano

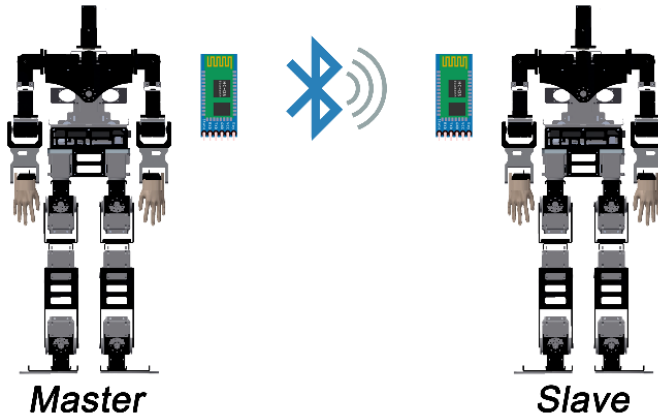
### 3.3. Desain Komunikasi Antar Robot

Pada Kontes Robot Seni Tari Indonesia, robot yang digunakan berjumlah dua unit. Maka dari itu, dirancanglah komunikasi antara dua robot agar dapat serasi dalam melakukan setiap gerakan. Dua robot tersebut kita sebut master dan slave. Pada perancangan elektronik robot slave tidak memiliki beberapa komponen elektronik seperti tidak adanya sensor suara, karena semua kendali pada robot master. Sistem komunikasi sendiri menggunakan Bluetooth HC-05 yang sudah dihubungkan pair. Skema komunikasi antar robot seperti pada gambar 3.9.

Pemilihan Bluetooth HC-05 sebagai perantara pengiriman data dikarenakan proses pairing cepat lebih aman, ditambah adanya led indikator pada saat proses pairing berlangsung. Bluetooth HC-05 sendiri memanfaatkan gelombang radio berfrekuensi 2,4 GHz sehingga jangkauannya sendiri bias sampai 10 meter menurut datasheet.

Cara kerja Bluetooth ini sendiri adaah sebagai berikut. Bluetooth master akan mengirimkan perintah ke Bluetooth slave apabila ada

musik atau tidak, sehingga robot dapat menari dengan serasi. Data yang dikirimkan hanya berupa karakter “1” ketika ada musik dan “0” ketika tidak ada musik. Berikut adalah ilustrasi komunikasi antar robot.



**Gambar 3.9.** Komunikasi Antar Robot.

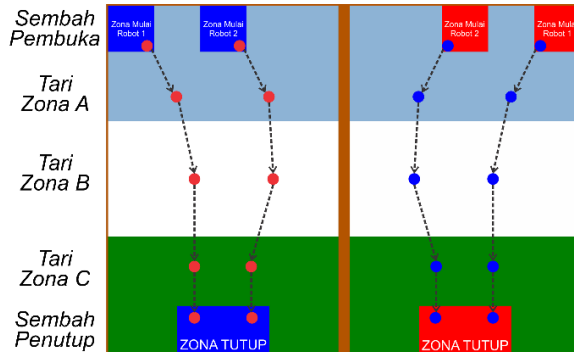
### **3.4. Perancangan Gerakan Tari Remo**

Karakteristik pada tata gerak Tari Remo adalah gerakan kaki yang rancak dan dinamis. Gerakan ini didukung dengan adanya bandul-bandul (binggel) yang dipasang di pergelangan kaki. Bandul lonceng ini berbunyi saat penari melangkah atau menghentakkan kakinya di panggung. Selain itu, ciri khas yang lain adalah gerakan melempar selendang atau sumpur secara cepat dan dinamis, gerakan anggukan dan gelengan kepala, ekspresi wajah, serta kuda-kuda penari membuat tarian ini menjadi semakin atraktif. [1]

Gerakan robot akan dibagi menjadi 3 zona, yaitu zona A, zona B, dan zona C. Pada zona A berukuran 1000x2000 mm terdapat tempat zona mulai untuk robot pertama dan robot kedua dengan ukuran masing-masing 400x400 mm. Pada zona A robot dapat melakukan gerakan sembah pembuka, tindak gedrug, gedrug rangkep, keter dan iket. Pada zona B robot dapat melakukan gerakan Gerak sabetan, bumi langit, keter dan iket. Pada zona C terdapat juga Zona tutup yang digunakan untuk robot mengakhiri tarian. Di zona C robot dapat melakukan gerakan Pentangan kanan kiri buang sumpur, pentangan onkean lambung, numpang tali dan tanjak tancep

bersamaan dengan sembah penutup.

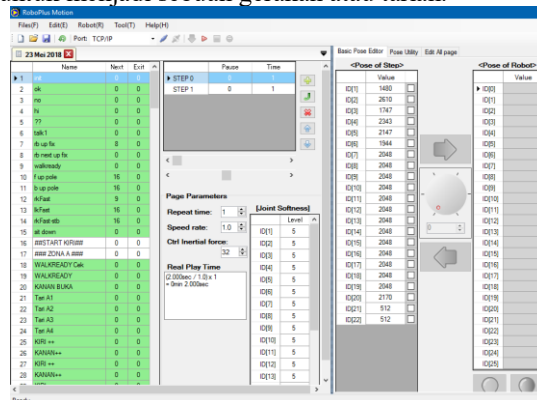
Lapangan pertandingan yang digunakan berbentuk segi empat dengan ukuran 2000x3000 mm. Lapangan pertandingan dan pola lantai seperti ditunjukkan gambar 3.10.



**Gambar 3.10.** Pola Lantai yang Diterapkan

### 3.4.1 RoboPlus Motion

Software RoboPlus Motion merupakan software bawaan dari Robotis untuk membuat gerakan tiap frame pada robot. Selain itu software ini juga dapat menyimpan seluruh frame gerakan yang telah diatur yang selanjutnya akan dipanggil oleh program pada main controller untuk menjadi sebuah gerakan atau tarian.



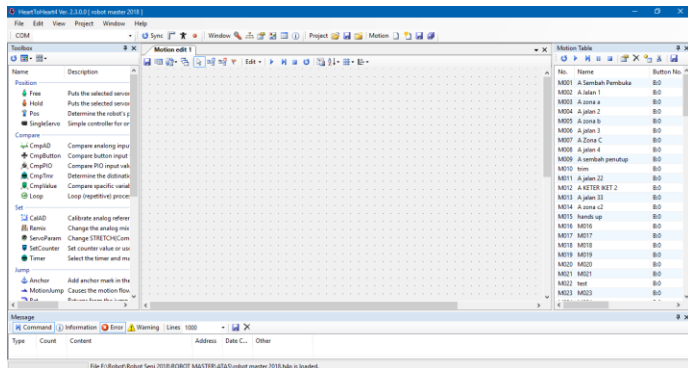
**Gambar 3.11.** Software RoboPlus Motion

Tampilan software RoboPlus Motion seperti gambar 3.11.

Penggunaan RoboPlus Motion ini sendiri hanya untuk servo dari Dynamixel, sehingga hanya untuk menggerakkan servo bagian bawah.

### 3.4.2 HeartToHeart4

Software HeartToHeart4 merupakan software bawaan dari Kondo Kagaku untuk membuat sebuah project motion dan beberapa perintah input/output yang selanjutnya akan disimpan di servo *controller* RCB-4HV. Pada penggunaannya, *software* ini hanya diperuntukkan untuk servo bagian atas. Tampilan *software* HeartToHeart4 seperti gambar 3.12. Software ini selain untuk membuat beberapa motion juga dapat membuat beberapa fungsi seperti fungsi Loop, fungsi Timer, fungsi delay dan sebagainya yang selanjutnya akan di upload di RCB-4HV.



**Gambar 3.12.** Software HeartToHeart4

## 3.5. Implementasi Desain Mekanik Robot

Setelah proses desain mekanik selesai, selanjutnya akan melalui proses fabrikasi part alumunium menggunakan beberapa mesin pendukung seperti mesin laser cutting, mesin bending dan mesin welding. Beberapa part dibuat menggunakan mesin 3D Print seperti pada bagian telapak tangan. Proses fabrikasi part ini dilakukan di perusahaan Dempo Metal Laserindo, Rungkut, Surabaya. Seluruh part hasil fabrikasi selanjutnya dirakit menjadi robot.

Dalam proses perakitan sendiri ada beberapa part yang harus

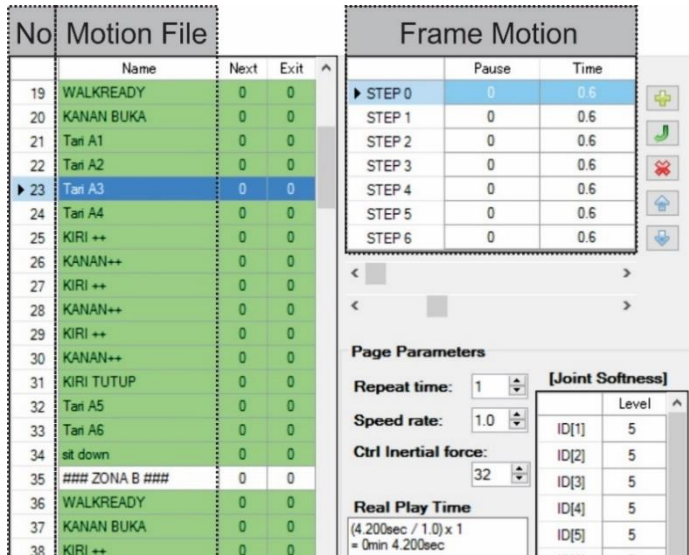
dimodifikasi seperti penambahan ulir menggunakan handtap. Servo Dynamixel MX-28T yang akan dipasangkan ke part aluminium harus dikalibrasi dan diatur titik tengahnya. Proses perakitan robot humanoid ini sendiri menghabiskan waktu 7 hari, karena ada beberapa part yang harus dikembalikan ke pihak fabrikasi. Desain robot humanoid ini bisa dilihat pada gambar 3.13.



**Gambar 3.13.** Desain robot lengkap dengan aksesoris

### **3.6. Pembuatan Gerakan Tari Remo**

Untuk membuat frame gerakan pada robot kita menggunakan dua software, yaitu RoboPlus Motion untuk servo jenis Dynamixel MX-28T dan HeartToHeart4 untuk servo jenis Kondo. Pada software RoboPlus Motion, kita dapat membuat maksimal 7 frame dalam 1 motion file. Motion file ini nantinya akan dipanggil oleh program dari main controller sesuai dengan nomor motion file tersebut.



**Gambar 3.14.** Software Roboplus

Dari gambar 3.14 kita dapat mengetahui motion file “Tari A3” dengan nomor “23” memiliki 7 frame motion. Berikut adalah contoh program untuk memanggil file tersebut :

```

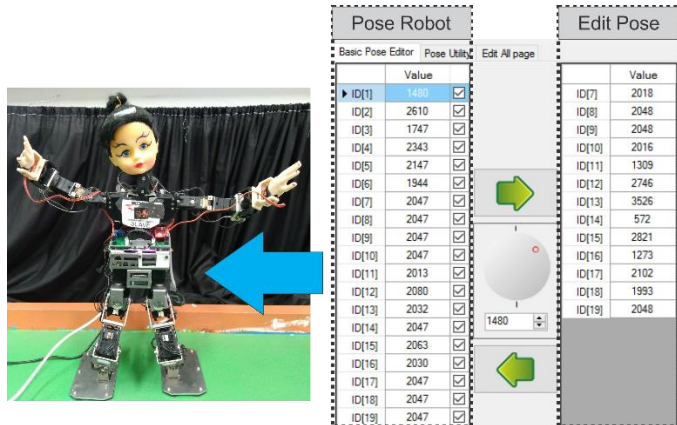
else if (motion_state == 2) {
    motion_state = 3;
    Action::GetInstance()->Start(20);
    while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
else if (motion_state == 3) {
    motion_state = 4;
    Action::GetInstance()->Start(21);
    while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
else if (motion_state == 4) {
    motion_state = 5;
    Action::GetInstance()->Start(22);
    while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
else if (motion_state == 5) {
    motion_state = 6;
    Action::GetInstance()->Start(23);
    while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}

```

**Gambar 3.15.** Program memanggil motion

Tiap frame pada roboplus memiliki nilai posisi motor pada setiap joint sehingga bisa didapatkan pose pada robot. Ketika nilai posisi

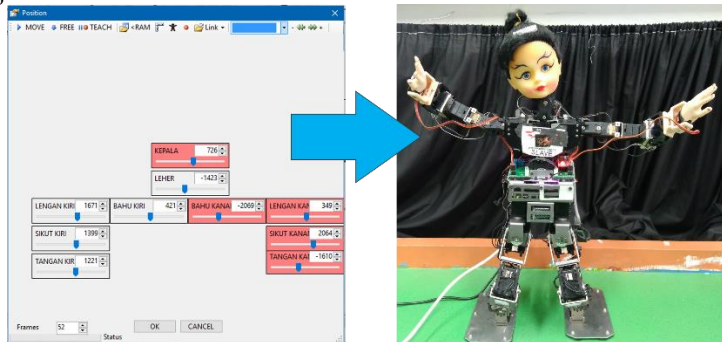
motor pada frame selanjutnya berubah, pose pada robot juga akan berubah.



**Gambar 3.16.** Pose Robot dengan RoboPlus Motion

Pada gambar 3.16, nilai pose robot adalah nilai seluruh motor yang sudah dikumpulkan di main controller, sedangkan nilai edit pose adalah nilai yang dapat dimodifikasi yang selanjutnya bisa di upload di nilai pose robot. Pada servo jenis MX-28T memiliki nilai resolusi 4096 atau 360° sehingga nilai titik tengahnya 2048 (90°).

Untuk mengatur servo bagian atas yang menggunakan servo jenis Kondo memakai software HeartToHeart4.



**Gambar 3.17.** Pose Robot dengan HeartToHeart4

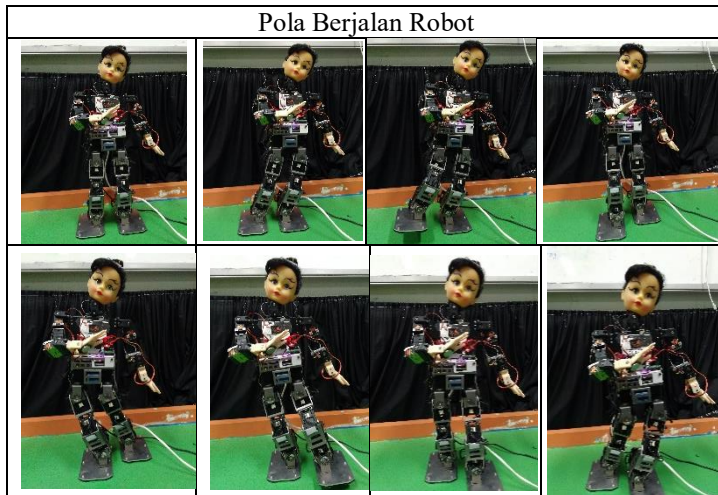
Pada platform kondo ini memiliki nilai resolusi antara 0 sampai 15000 (360°), sehingga nilai titik tengah servo didapatkan nilai 7500

(90°). Di dalam software HeartToHeart4 ini sendiri memiliki beberapa program fungsi yang dapat dijalankan bersamaan dengan motion sehingga kita tidak lagi membuat program di main controller. Seluruh project motion yang dibuat selanjutnya akan diupload ke Servo Controller RCB-4HV melalui komunikasi serial.

### 3.7. Implementasi Gerakan Tari Remo

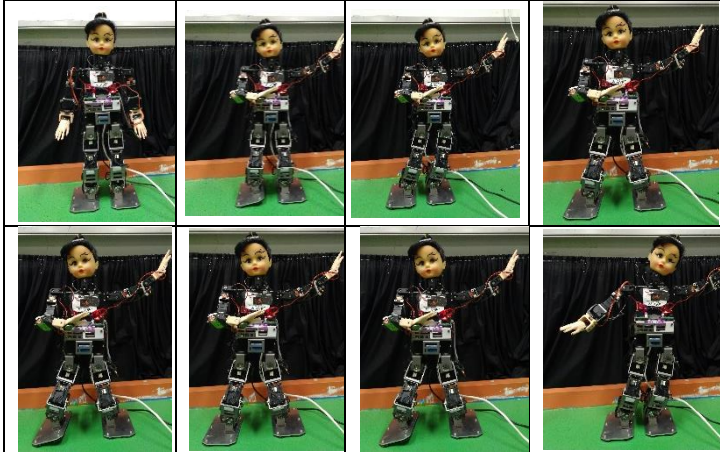
Pada tari remo, gerakan dasar pada bagian kaki terdiri dari *Jalan*, *Gejutan*, *sabetan* dan *tanjak tancep*. Gerakan dasar jalan sama seperti manusia pada umumnya, kaki kiri dan kaki kanan secara bergantian melangkah. *Gejutan* adalah posisi kaki dalam kondisi kuda-kuda sedangkan kaki kiri atau kanan melakukan *nggejug* ke lantai beberapa kali. Sedangkan *Sabetan* adalah posisi dimana kaki sebelah kanan diangkat setinggi lutut lalu melakukan *sabetan* pada selendang. *Tanjak tancep* merupakan gerakan pada zona C yang dibuat sedemikian rupa seperti menggoyangkan badan ke kiri dan ke kanan selanjutnya mengangkat sampur ke atas dan kebawah secara bergantian. Berikut adalah tabel gerakan pola berjalan robot dengan model tangan kanan dan tangan kiri berat ke samping kiri dan beberapa gerakan tarian remo seperti *gejukan*, *sabetan*, dan *tanjak tancep*.

**Tabel 3.2.** Gerakan Dasar Tari Remo

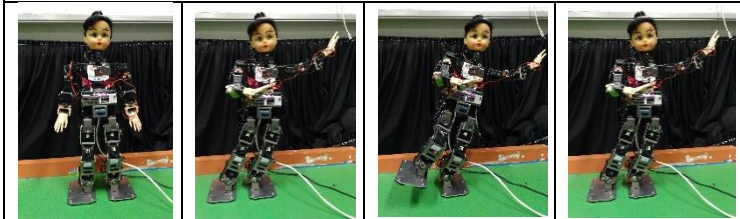




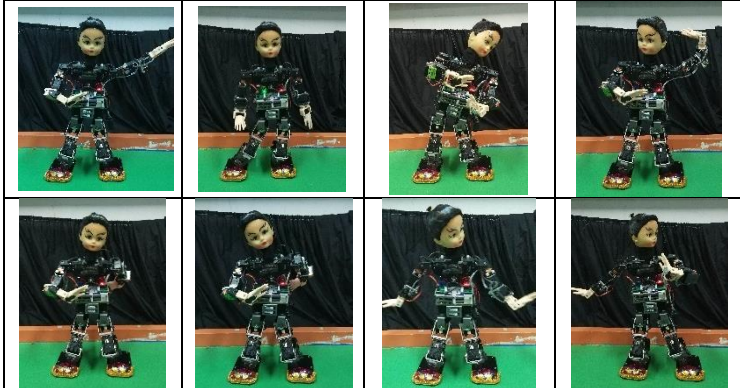
### Gerakan Gejungan



### Gerakan Sabetan



### Gerakan Tanjak Tancep



### 3.8. Sistem Penilaian KRSTI 2018

Sistem penilaian KRSTI 2018 sendiri mengacu pada buku panduan yang ditetapkan oleh juri dari kemenristek dikti yang meliputi beberapa kategori seperti pada tabel berikut :

**Tabel 3.3.** Penilaian KRSTI 2018

Nama	Keterangan	Point	Jumlah
Zona Mulai	Robot mampu melakukan gerakan Sembah Pembuka	1 – 10	
Zona A	Tindak, Gedrug, Gedrug Rangkep, Keter & Iket	1 – 10	
Zona B	Gerak Sabetan, Bumi Langit, Keter & Iket	1 – 10	
Zona C	Pentangan Kanan Kiri kanan buang sampur, ongkean lambung, tumpang tali, tanjak tancep	1 – 10	
Zona Tutup	Tanjak Tancep Sembahan	1 – 10	
Point Tambahan	Sinkronisasi Gerakan	1 – 10	
	Robot mencapai Zona A	5	
	Robot mencapai Zona B	5	
	Robot mencapai Zona C	5	
	Robot mencapai Zona Tutup	5	
	Keindahan gerak dan selaras dengan tari remo (Faktor pengali 2)	(1 – 10) x 2	
Total Point			

Selain nilai penampilan, ada beberapa peraturan apabila robot error yang mengharuskan untuk meminta Retry. Retry sendiri akan menghilangkan nilai bonus dimana robot error pada zona tersebut.

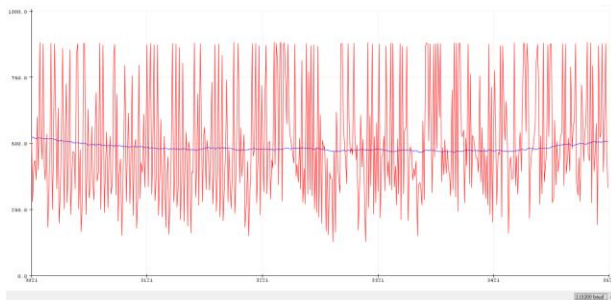
## **BAB IV**

### **PENGUJIAN DAN ANALISIS**

Bab ini membahas tentang pengujian dari sistem yang telah dirancang beserta analisis data yang diperoleh saat pengujian. Pengujian-pengujian yang dilakukan meliputi pengujian sensor suara, pengujian komunikasi antar robot dan pengujian tarian remo pada robot. Robot humanoid virose ini juga akan melakukan pengujian pada seleksi tingkat regional di Politeknik Negeri Malang pada tanggal 1 -3 Mei 2018 dan apabila lolos, akan lanjut ke tingkat nasional yang akan diselenggarakan di Universitas Negeri Yogyakarta pada tanggal 11 – 14 Juli 2018.

#### **4.1. Pengujian Sensor Suara**

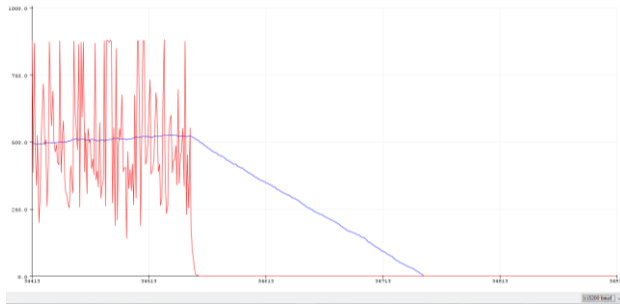
Pengujian sensor suara sebelum di implementasikan ke robot perlu dilakukan supaya didapatkan nilai threshold dan respon dari sensor suara tersebut. Diharapkan dengan adanya pengujian ini robot dapat merespon adanya suara yang ditransferkan oleh juri melalui Bluetooth audio. Berikut grafik dari respon sensor suara yang digunakan :



**Gambar 4.1.** Grafik Output Sensor Suara.

Output dari Bluetooth Audio Receiver dimasukkan kedalam sensor suara karena outputnya terlalu kecil untuk dibaca oleh sub controller (Arduino Nano) melalui pin Analog Digital Converter (ADC), range output tersebut sekitar -0,7v sampai 0,7v. Maka perlu dilakukan penguatan yang dilakukan oleh sensor suara. Output sensor suara dirasa masih terlalu banyak noise, sehingga dilakukan filter melalui suatu program. Pada gambar 4.1, grafik berwarna merah

merupakan hasil output yang belum di filter, sedangkan grafik warna biru hasil output yang telah melalui filter digital.



**Gambar 4.2.** Grafik Sensor Suara ketika di Mute.

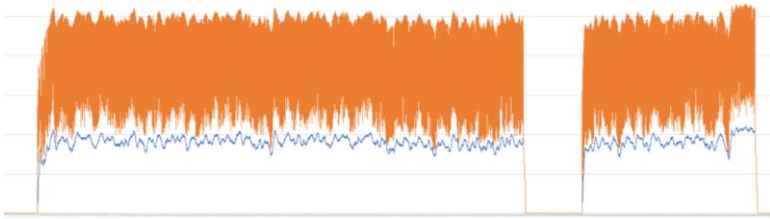
Sedangkan gambar 4.2 merupakan kondisi ketika lagu di mute. Berikut adalah program filter yang diterapkan pada robot.

```
float filter(float data)
{
    float out;
    int jumlah_data = 199;
    static float filter_array[200];
    float sum = 0;

    for(int i = jumlah_data-1 ; i > 0; i--)
    {
        filter_array[i] = filter_array[i-1];
    }
    filter_array[0] = data;
    for(int i = 0 ; i < jumlah_data; i++)
    {
        sum += filter_array[i];
    }
    out = sum / jumlah_data;
    return out;
}
```

Hasil pembacaan sensor suara dapat dikatakan berhasil tanpa adanya nilai floating maupun ripple, sehingga nilai threshold bisa diatur pada posisi nilai 50. Secara keseluruhan, hasil pembacaan

sensor dapat dilihat pada gambar 4.3.



**Gambar 4.3.** Pembacaan Sensor Suara.

#### 4.2. Pengujian Komunikasi Main Controller dengan Sub Controller

Pengujian ini bertujuan untuk melihat respon pada robot ketika pengiriman data dari sensor suara ke main controller. Robot akan berhenti ketika sub controller Arduino Nano mengirim data serial character 'A' dan CM-730 mengirim nilai  $ADC < 200$ . Apabila main controller menerima data serial 'B' atau CM-730 mengirim nilai  $ADC \geq 200$  maka robot akan melakukan gerakan. Berikut contoh tabel kebenaran yang diterapkan pada robot humanoid Virose.

**Tabel 4.1.** Truth Table pada Robot Virose

Input		Output
Serial Data	ADC	
'A'	$<200$	STOP
'A'	$\geq 200$	GO
'B'	$<200$	GO
'B'	$\geq 200$	GO

Pada pengujian ini robot berhasil melakukan simulasi sesuai dengan truth table. Hasil monitoring robot dapat dilihat pada gambar 4.4 saat keadaan 'STOP' dan gambar 4.5 saat keadaan 'GO'.



pada sistem penilaian KRSTI 2018 yang berupa tiga kali penilaian tahap awal, dua kali pertandingan 8 besar dan dua kali pertandingan final. Sistem penilaian ini dilakukan dengan bentuk centang bukan point. Berikut adalah hasil dari pengujiannya.

**Tabel 4.2.** Hasil Pengujian di Lapangan Latihan

Nama	Keterangan	Percobaan						
		1	2	3	4	5	6	7
Zona Mulai	Robot mampu melakukan gerakan Sembah Pembuka	✓	✓	✓	✓	✓	✓	✓
Zona A	Tindak, Gedrug, Gedrug Rangkep, Keter & Iket	✓	✓	✓	✓	✓	✓	✓
Zona B	Gerak Sabetan, Bumi Langit, Keter & Iket	✓	✓	✓	✓	✓	✓	✓
Zona C	Pentangan Kanan Kiri kanan buang sampur, ongkean lambung, tumpang tali, tanjak tancep	✓	✓	✓	✓	✓	✓	✓
Zona Tutup	Tanjak Tancep Sembahan	✓	✓	✓	✓	✓	✓	✓
Point Tambahan	Sinkronisasi Gerakan	✗	✓	✓	✓	✓	✓	✓
	Robot mencapai Zona A	✓	✓	✓	✓	✓	✓	✓
	Robot mencapai Zona B	-	-	-	✓	✓	✓	✓
	Robot mencapai Zona C	✓	✓	✓	✓	✓	✓	✓
	Robot mencapai Zona Tutup	✓	✓	✓	✓	✓	✓	✓
	Keindahan gerak dan selaras dengan tari remo	✓	✓	✓	✓	✓	✓	✓

	(Faktor pengali 2)						
--	-----------------------	--	--	--	--	--	--

Dari hasil pengujian di lapangan latihan didapatkan error saat percobaan pertama pada Sinkronasi Gerakan. Hal ini dikarenakan Bluetooth master dan slave belum pairing saat start, namun hal ini nantinya dapat diminimalisir dengan proses pairing lebih awal. Selain itu robot juga tidak mendapatkan point tambahan pada percobaan 1, 2 dan 3. Hal ini dikarenakan robot meminta retry sehingga nilai bonus pada zona tersebut dihilangkan.

#### 4.3.2. Pengujian Robot di KRSTI tingkat Regional

Pengujian ini bertujuan untuk mengetahui performa robot dari segi gerakan, stabilitas, dan komunikasi. Pengujian ini dilakukan di Politeknik Negeri Malang pada tanggal 1 – 3 Mei 2018. Berikut adalah tabel nilai yang diperoleh dari dewan juri.

**Tabel 4.3.** Hasil Nilai tingkat Regional

Nama Tim	Perguruan Tinggi	P1	P2	P3	Nilai
ERISA	PENS	81.67	77.67	79.00	79.44
Azzahrally	UNESA	73.33	67.67	72.00	71.00
Nakula Nayaka	UB	73.67	66.67	71.67	70.67
<b>VI-ROSE</b>	<b>ITS</b>	<b>77.67</b>	<b>42.33</b>	<b>46.00</b>	<b>55.33</b>
e-Pretiwi	ITN Malang	45.67	48.00	51.00	48.22
AERORA	UN Malang	54.00	44.33	41.00	46.44
Putra Mandalika	Univ Mataram	16.67	27.00	22.33	22.00
BlueDance6	STMIK Tarakanita	19.67	9.33	19.67	16.22
Ashuro	Unair	8.00	16.00	13.00	12.33
COS	Univ Merdeka Malang	6.33	15.67	12.33	11.44
IRAGROTIK'64	Poltek Jember	2.67	19.00	11.00	10.89
ASCARYA	Univ Udayana	7.00	6.00	14.00	9.00
W-ONE	Univ Kanjuruhan Malang	1.00	9.33	8.33	6.22
OCEANOID	Univ Hang Tuah	6.33	2.00	3.67	4.00

Dari hasil tingkat regional, performa robot pada penampilan pertama mendapatkan 77,67 point. Ini adalah kondisi dimana dua



robot virose tidak mengalami masalah. Sedangkan pada pertandingan kedua dan ketiga, robot master mengalami error pada bagian komunikasi main controller dengan sub controller CM-730. Oleh karena itu, perbaikan demi perbaikan untuk mengatasi masalah tersebut didapatkan solusi seperti yang dijelaskan pada bab 4.2.

#### 4.3.3. Pengujian Robot di KRSTI tingkat Nasional

Pengujian ini berlanjut di tingkat Nasional yang dilakukan di Universitas Muhammadiyah Yogyakarta pada tanggal 11 – 14 Juli 2018. Berikut adalah tabel nilai yang diperoleh dari dewan juri.

**Tabel 4.4.** Hasil Nilai 14 Besar tingkat Nasional

Nama Tim	Perguruan Tinggi	P1	P2	P3	Nilai
Rosemery	UNY	84.67	73.67	75.33	77.89
Azzahrally	UNESA	76.33	77.67	77.00	77.00
ERISA	PENS	77.00	77.00	76.00	76.67
<b>VI-ROSE</b>	<b>ITS</b>	<b>75.00</b>	<b>75.67</b>	<b>73.00</b>	<b>74.56</b>
Lanange Jagad	UAD	70.33	73.00	78.00	73.78
CMCS ART	PNUP	73.67	70.00	72.67	72.11
Dago Concordia	ITB	70.67	76.33	64.67	70.56
Nakula Nayaka	UB	71.00	69.67	71.00	70.56
Badaya SAS	Telkom Univ	66.00	66.67	70.00	67.56
Euro Arte	USR	67.67	62.33	68.00	66.00
ALFAN	UGM	58.67	66.00	68.67	64.44
Robogen	STMIK Lampung	53.67	65.67	58.33	59.22
Jan'nah	UNNES	40.00	60.33	69.00	56.44
Pagaruyung	UNP	48.00	59.00	55.33	54.11

**Tabel 4.5.** Hasil Nilai 8 Besar tingkat Nasional

Nama Tim	Perguruan Tinggi	Nilai
Rosemery	UNY	91.00
ERISA	PENS	80.67
<b>VI-ROSE</b>	<b>ITS</b>	<b>80.00</b>
Azzahrally	UNESA	78.33
Nakula Nayaka	UB	78.00
Lanange Jagad	UAD	76.00

CMCS ART	PNUP	75.67
Dago Concordia	ITB	71.33

**Tabel 4.6.** Hasil Nilai 4 Besar tingkat Nasional

Nama Tim	Perguruan Tinggi	Nilai
Rosemery	UNY	91.00
<b>VI-ROSE</b>	<b>ITS</b>	<b>82.00</b>
ERISA	PENS	72.00
Azzahrally	UNESA	70.00

Dari hasil tabel 4.4, tabel 4.5, dan tabel 4.6 penampilan robot ini mengalami peningkatan. Sehingga robot ini layak mendapatkan juara 2 di tingkat nasional pada ajang KRSTI 2018.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan percobaan yang telah dilakukan pada pelaksanaan tugas akhir ini didapat beberapa kesimpulan sebagai berikut:

1. Desain mekain robot humanoid virose ini berhasil diimplementasikan, robot dapat melakukan gerakan tari remo dengan rancak.
2. Desain elektronik dan komunikasi pada robot sudah dapat diimplementasikan pada robot. Dengan penambahan beberapa kemungkinan yang mungkin bisa saja terjadi.
3. Dari hasil regional didapatkan bahwa robot humanoid virose ini bisa bersaing dengan tim-tim lain, hal ini dibuktikan dengan hasilnya Juara 4 dan lolos ke tingkat nasional.
4. Pengujian di tingkat Nasional, robot dapat melakukan gerakan tari remo dengan baik, dengan mendapatkan juara 2 tingkat nasional.

#### **5.2. Saran**

Sebagai sarana pengembangan robot humanoid ini, maka terdapat beberapa saran dari penulis berdasarkan hasil yang diperoleh saat percobaan, yaitu sebagai berikut:

1. Penyempurnaan desain mekanik dengan penambahan jumlah DoF, agar didapatkan gerakan yang lebih banyak.
2. Penerapan sistem keseimbangan pada robot.
3. Penggunaan mini PC sebagai main controller dirasa terlalu mengganggu, sehingga penampilan robot terkesan kaku pada bagian pinggul. Selain itu penggantian mini PC juga perlu dilakukan kedepannya dengan menggunakan kontrolller sendiri.

.....*Halaman ini sengaja dikosongkan*.....

## DAFTAR PUSTAKA

- [1] Joseph Rudy, "Zero Moment Point Walking Controller for Humanoid Walking using Drwin-OP," University of Notre Dame, Indiana, 2014.
- [2] Direktorat Jenderal Pembelajaran dan Kemahasiswaan Kementrian Riset, Teknologi, dan Pendidikan Tinggi, "Kontes Robot Seni Tari Indonesia 2018," in *Buku Panduan KRSTI 2018*, Panitia Kontes Robot Seni Tari Indonesia 2018, p. 5.
- [3] I. Ha, Y. Tamura, and H. Asama, "Development of open platform humanoid robot DARwIn-OP," vol. 27, pp. 223–232, Feb. 2013.
- [4] "<http://support.robotis.com/en/product/darwin-op.htm>," Mei-2018. .
- [5] "<https://kondo-robot.com/product/khr-3hv-ver-2-life>," *KHR-3HV Ver.2*, 15-Jul-2018. .
- [6] "<https://components101.com/microcontrollers/arduino-nano>," Mei-2018. .
- [7] A. D. Anggono, Bibit Sugito, Agus Hariyanto, and Subroto, "Mechanical Behaviour Investigation Of Aluminium Alloy Tailor Welded Blank Developed By Using Friction Stir Welding Technique," p. 7.

.....*Halaman ini sengaja dikosongkan*.....

## **LAMPIRAN A**

### ***Program pada Arduino***

```
#include <SoftwareSerial.h>
#include <Servo.h>

SoftwareSerial BTserial(5, 3); //tx,rx
Servo servokanan;
Servo servokiri;
int a0; //sensor suara
int state = 0;
int pos = 0;
bool maju = true;
int threshold = 75;

int LedStateSS = 4; //led status SS
int rcb = 2; // data ke rcb

//input ke servo
int input1 = 7; //input1
int input2 = 8; //input2
int input3 = 12; //input3
int led = 13;
int A = 0;
int B = 0;
int C = 0;

//panic button
int analogPin = 2; //input state
int panicbutton = 0;

//millis
long previousMillis = 0; // akan menyimpan milidetik terakhir dari loop
long interval;           // interval blink dalam ms pengganti delay
unsigned long currentMillis;

//gpio ke CM
int cm = 10;
```

```

void setup() {
    Serial.begin(115200);
    BTserial.begin(38400);
    pinMode(led, OUTPUT); //Pin Led
    pinMode(LedStateSS, OUTPUT); //LED Status
    pinMode(rcb, OUTPUT);
    pinMode(cm, OUTPUT); //GPIO CM
    digitalWrite(rcb, LOW); //state awal rcb
    digitalWrite(cm, LOW); //state awal cm

    servokanan.attach(6);
    servokiri.attach(9);
    servokiri.write(90);
    servokanan.write(90);

    pinMode(input1, INPUT); //GPIO Servo
    pinMode(input2, INPUT); //GPIO Servo
    //pinMode(input3, INPUT); //GPIO Servo
}

float filter(float data)
{
    float out;
    int jumlah_data = 199;
    static float filter_array[200];
    float sum = 0;

    for(int i = jumlah_data-1 ; i > 0; i--)
    {
        filter_array[i] = filter_array[i-1];
    }

    filter_array[0] = data;

    for(int i = 0 ; i < jumlah_data; i++)
    {
        sum += filter_array[i];
    }
}

```



```

    out = sum / jumlah_data;

    return out;
}

void timer(){
    if(currentMillis - previousMillis > interval) { // kondisi untuk
menghitung apakah interval > dari jumlah milisecond saat ini
        previousMillis = currentMillis;
    }
}

void loop()
{
    int SensorSuara = analogRead(A0); //Membaca nilai dari Sensor Suara
    SensorSuara = filter((float)SensorSuara);
    // Serial.print("\n filter = ");
    // Serial.print(SensorSuara);
    // Serial.print("\t no filter = ");
    // Serial.print(analogRead(A0));
    // Serial.print("\t");

    A = (digitalRead(input1));
    B = (digitalRead(input2));

    panicbutton = analogRead(analogPin);

    // jumlah milidetik saat ini, bertambah 1 tiap 1 loop
    currentMillis = millis();

    if (SensorSuara > threshold)
    {
        Serial.print("B");
        BTserial.write('1');//sends a 1 through the bluetooth serial link
        digitalWrite(rcb, HIGH);
        digitalWrite(cm, HIGH);
        digitalWrite(led, HIGH);
        digitalWrite(LedStateSS, HIGH);
    }
}

```

```

if (A==HIGH){
  if (B==HIGH){
    servokiri.write(90) ;
    servokanan.write(90);
    interval = 630;
    timer;
  }
  else {
    servokiri.write(0) ;
    servokanan.write(180);
    interval = 630;
    timer;
  }
}
else{
  if (B==HIGH){
    servokiri.write(90) ;
    servokanan.write(180);
    interval = 250;
    timer;
  }
  else{
    servokiri.write(90);
    servokanan.write(90);
    interval = 200;
    timer;
  }
}
}

else if (panicbutton > 750){
  A = (digitalRead(input1));
  B = (digitalRead(input2));
  C = (digitalRead(input3));
  Serial.print("B");
  BTserial.write('2');//sends a 2 through the bluetooth serial link
  digitalWrite(rcb, HIGH);
  digitalWrite(cm, HIGH); //CM ON
}

```

```

digitalWrite(led, LOW);
digitalWrite(LedStateSS, LOW);

if (A == HIGH && B == HIGH){
    servokiri.write(90) ;
    servokanan.write(90);
    interval = 250;
    timer;
}
else if (A == HIGH && B == LOW) {
    servokiri.write(0) ;
    servokanan.write(180);
    interval = 250;
    timer;
}
else if (A == LOW && B == HIGH){
    servokiri.write(90) ;
    servokanan.write(180);
    interval = 250;
    timer;
}
else{
    servokiri.write(90);
    servokanan.write(90);
    interval = 200;
    timer;
}
}
else {
    Serial.print("A");
    BTserial.write('0');
    digitalWrite(rcb, LOW);
    digitalWrite(cm, LOW);
    digitalWrite(led, LOW);
    digitalWrite(LedStateSS, LOW);

    if (A == HIGH && B == LOW) {
        servokiri.write(20) ;
        servokanan.write(180);
    }
}

```

```
        interval = 250;  
        timer;  
    }  
}
```

## LAMPIRAN B

### Program pada Sub Controller CM-730

```
#include <stdio.h>
#include "FSR.h"
#include "CM730.h"
#include "MotionStatus.h"

using namespace Robot;

#define ID (2)
#define LENGTH (3)
#define INSTRUCTION (4)
#define ERROBIT (4)
#define PARAMETER (5)
#define DEFAULT_BAUDNUMBER (1)

#define INST_PING (1)
#define INST_READ (2)
#define INST_WRITE (3)
#define INST_REG_WRITE (4)
#define INST_ACTION (5)
#define INST_RESET (6)
#define INST_SYNC_WRITE (131) // 0x83
#define INST_BULK_READ (146) // 0x92

BulkReadData::BulkReadData() :
    start_address(0),
    length(0),
    error(-1)
{
    for(int i = 0; i < MX28::MAXNUM_ADDRESS; i++)
        table[i] = 0;
}

int BulkReadData::ReadByte(int address)
{

```

```

        if(address >= start_address && address < (start_address + length))
            return (int)table[address];

        return 0;
    }

int BulkReadData::ReadWord(int address)
{
    if(address >= start_address && address < (start_address + length))
        return CM730::MakeWord(table[address], table[address+1]);

    return 0;
}

CM730::CM730(PlatformCM730 *platform)
{
    m_Platform = platform;
    DEBUG_PRINT = false;
    m_BulkReadTxPacket[LENGTH] = 0;
    for(int i = 0; i < ID_BROADCAST; i++)
        m_BulkReadData[i] = BulkReadData();
}

CM730::~~CM730()
{
    Disconnect();
}

int CM730::TxRxPacket(unsigned char *txpacket, unsigned char
*rxpacket, int priority)
{
    if(priority > 1)
        m_Platform->LowPriorityWait();
    if(priority > 0)
        m_Platform->MidPriorityWait();
    m_Platform->HighPriorityWait();

    int res = TX_FAIL;
    int length = txpacket[LENGTH] + 4;

```

```

txpacket[0] = 0xFF;
txpacket[1] = 0xFF;
txpacket[length - 1] = CalculateChecksum(txpacket);

if(DEBUG_PRINT == true)
{
    fprintf(stderr, "\nTX: ");
    for(int n=0; n<length; n++)
        fprintf(stderr, "%.2X ", txpacket[n]);

    fprintf(stderr, "\nINST: ");
    switch(txpacket[INSTRUCTION])
    {
    case INST_PING:
        fprintf(stderr, "PING\n");
        break;

    case INST_READ:
        fprintf(stderr, "READ\n");
        break;

    case INST_WRITE:
        fprintf(stderr, "WRITE\n");
        break;

    case INST_REG_WRITE:
        fprintf(stderr, "REG_WRITE\n");
        break;

    case INST_ACTION:
        fprintf(stderr, "ACTION\n");
        break;

    case INST_RESET:
        fprintf(stderr, "RESET\n");
        break;

    case INST_SYNC_WRITE:

```

```

        fprintf(stderr, "SYNC_WRITE\n");
        break;

    case INST_BULK_READ:
        fprintf(stderr, "BULK_READ\n");
        break;

    default:
        fprintf(stderr, "UNKNOWN\n");
        break;
    }
}

bool CM730::Connect()
{
    if(m_Platform->OpenPort() == false)
    {
        fprintf(stderr, "\n Fail to open port\n");
        fprintf(stderr, " CM-730 is used by another program or do not
have root privileges.\n\n");
        return false;
    }

    return DXLPowerOn();
}

bool CM730::ChangeBaud(int baud)
{
    if(m_Platform->SetBaud(baud) == false)
    {
        fprintf(stderr, "\n Fail to change baudrate\n");
        return false;
    }
    return DXLPowerOn();
}

bool CM730::DXLPowerOn()
{
    if(WriteByte(CM730::ID_CM, CM730::P_DXL_POWER,1,0) ==
CM730::SUCCESS)
    {

```



```

        if(DEBUG_PRINT == true)
            fprintf(stderr, " Succeed to change Dynamixel
power!\n");

        WriteWord(CM730::ID_CM, CM730::P_LED_HEAD_L,
MakeColor(255, 128, 0), 0);
        m_Platform->Sleep(300); // about 300msec
    }
    else
    {
        if(DEBUG_PRINT == true)
            fprintf(stderr, " Fail to change Dynamixel power!\n");
        return false;
    }
    return true;
}

void CM730::Disconnect()
{
    // Make the Head LED to green
    //WriteWord(CM730::ID_CM, CM730::P_LED_HEAD_L,
MakeColor(0, 255, 0), 0);
    unsigned char txpacket[] = {0xFF, 0xFF, 0xC8, 0x05, 0x03, 0x1A,
0xE0, 0x03, 0x32};
    m_Platform->WritePort(txpacket, 9);
    m_Platform->ClosePort();
}

int CM730::WriteByte(int address, int value, int *error)
{
    return WriteByte(ID_CM, address, value, error);
}

int CM730::WriteWord(int address, int value, int *error)
{
    return WriteWord(ID_CM, address, value, error);
}

int CM730::ReadByte(int id, int address, int *pValue, int *error)

```

```

{
    unsigned char txpacket[MAXNUM_TXPARAM + 10] = {0, };
    unsigned char rxpacket[MAXNUM_RXPARAM + 10] = {0, };
    int result;

    txpacket[ID] = (unsigned char)id;
    txpacket[INSTRUCTION] = INST_READ;
    txpacket[PARAMETER] = (unsigned char)address;
    txpacket[PARAMETER+1] = 1;
    txpacket[LENGTH] = 4;

    result = TxRxPacket(txpacket, rxpacket, 2);
    if(result == SUCCESS)
    {
        *pValue = (int)rxpacket[PARAMETER];
        if(error != 0)
            *error = (int)rxpacket[ERRBIT];
    }

    return result;
}
int CM730::ReadWord(int id, int address, int *pValue, int *error)
{
    unsigned char txpacket[MAXNUM_TXPARAM + 10] = {0, };
    unsigned char rxpacket[MAXNUM_RXPARAM + 10] = {0, };
    int result;

    txpacket[ID] = (unsigned char)id;
    txpacket[INSTRUCTION] = INST_READ;
    txpacket[PARAMETER] = (unsigned char)address;
    txpacket[PARAMETER+1] = 2;
    txpacket[LENGTH] = 4;

    result = TxRxPacket(txpacket, rxpacket, 2);
    if(result == SUCCESS)
    {
        *pValue = MakeWord((int)rxpacket[PARAMETER],
(int)rxpacket[PARAMETER + 1]);
    }
}

```

```

        if(error != 0)
            *error = (int)rxpacket[ERRBIT];
    }
    return result;
}

int CM730::ReadTable(int id, int start_addr, int end_addr, unsigned char
*table, int *error)
{
    unsigned char txpacket[MAXNUM_TXPARAM + 10] = {0, };
    unsigned char rxpacket[MAXNUM_RXPARAM + 10] = {0, };
    int result;
    int length = end_addr - start_addr + 1;

    txpacket[ID] = (unsigned char)id;
    txpacket[INSTRUCTION] = INST_READ;
    txpacket[PARAMETER] = (unsigned char)start_addr;
    txpacket[PARAMETER+1] = (unsigned char)length;
    txpacket[LENGTH] = 4;

    result = TxRxPacket(txpacket, rxpacket, 1);
    if(result == SUCCESS)
    {
        for(int i=0; i<length; i++)
            table[start_addr + i] = rxpacket[PARAMETER + i];
        if(error != 0)
            *error = (int)rxpacket[ERRBIT];
    }

    return result;
}

int CM730::WriteByte(int id, int address, int value, int *error)
{
    unsigned char txpacket[MAXNUM_TXPARAM + 10] = {0, };
    unsigned char rxpacket[MAXNUM_RXPARAM + 10] = {0, };
    int result;

    txpacket[ID] = (unsigned char)id;

```

```

txpacket[INSTRUCTION] = INST_WRITE;
txpacket[PARAMETER]    = (unsigned char)address;
txpacket[PARAMETER+1]  = (unsigned char)value;
txpacket[LENGTH]       = 4;

result = TxRxPacket(txpacket, rxpacket, 2);
if(result == SUCCESS && id != ID_BROADCAST)
{
    if(error != 0)
        *error = (int)rxpacket[ERRBIT];
}
return result;
}

int CM730::WriteWord(int id, int address, int value, int *error)
{
    unsigned char txpacket[MAXNUM_TXPARAM + 10] = {0, };
    unsigned char rxpacket[MAXNUM_RXPARAM + 10] = {0, };
    int result;

    txpacket[ID] = (unsigned char)id;
    txpacket[INSTRUCTION] = INST_WRITE;
    txpacket[PARAMETER]    = (unsigned char)address;
    txpacket[PARAMETER+1]  = (unsigned
char)GetLowByte(value);
    txpacket[PARAMETER+2]  = (unsigned
char)GetHighByte(value);
    txpacket[LENGTH]       = 5;
    result = TxRxPacket(txpacket, rxpacket, 2);
    if(result == SUCCESS && id != ID_BROADCAST)
    {
        if(error != 0)
            *error = (int)rxpacket[ERRBIT];
    }
    return result;
}

int CM730::MakeWord(int lowbyte, int highbyte)
{

```

```

        unsigned short word;
        word = highbyte;
        word = word << 8;
        word = word + lowbyte;
        return (int)word;
    }
int CM730::GetLowByte(int word)
{
    unsigned short temp;
    temp = word & 0xff;
    return (int)temp;
}
int CM730::GetHighByte(int word)
{
    unsigned short temp;
    temp = word & 0xff00;
    return (int)(temp >> 8);
}
int CM730::MakeColor(int red, int green, int blue)
{
    int r = red & 0xFF;
    int g = green & 0xFF;
    int b = blue & 0xFF;
    return (int)((((b>>3)<<10)|((g>>3)<<5)|(r>>3)));
}

```

## **LAMPIRAN C**

### **Program Interface**

```
#include <stdio.h>
#include <unistd.h>

#include "StatusCheck.h"
#include "Head.h"
#include "Action.h"
#include "Walking.h"
#include "MotionStatus.h"
#include "MotionManager.h"
#include "LinuxActionScript.h"

using namespace Robot;

int StatusCheck::m_cur_mode    = READY;
int StatusCheck::m_old_btn     = 0;
int StatusCheck::m_is_started  = 0;

void StatusCheck::Check(CM730 &cm730, int &jatuh)
{
    if(MotionStatus::FALLEN != STANDUP && (m_cur_mode ==
MOTION || m_cur_mode == SOCCER) && m_is_started == 1)
    {
        Walking::GetInstance()->Stop();
        while(Walking::GetInstance()->IsRunning() == 1)
        usleep(8000);

        Action::GetInstance()->m_Joint.SetEnableBody(true, true);
        jatuh = 1;
        while(Action::GetInstance()->IsRunning() == 1)
        usleep(8000);

        Head::GetInstance()->m_Joint.SetEnableHeadOnly(true, true);
        Walking::GetInstance()-
>m_Joint.SetEnableBodyWithoutHead(true, true);
    }
}
```

```

    if(m_old_btn == MotionStatus::BUTTON)
        return;

    m_old_btn = MotionStatus::BUTTON;

    if(m_old_btn & BTN_MODE)
    {
        fprintf(stderr, "Mode button pressed.. \n");

        if(m_is_started == 1)
        {
            m_is_started = 0;
            m_cur_mode = READY;
            LinuxActionScript::m_stop = 1;

            Walking::GetInstance()->Stop();
            Action::GetInstance()->m_Joint.SetEnableBody(true,
true);

            while(Action::GetInstance()->Start(15) == false)
                usleep(8000);
            while(Action::GetInstance()->IsRunning() == true)
                usleep(8000);
        }
        else
        {
            m_cur_mode++;
            if(m_cur_mode >= MAX_MODE) m_cur_mode =
READY;
        }
        MotionManager::GetInstance()->SetEnable(false);
        usleep(10000);
        if(m_cur_mode == READY)
        {
            cm730.WriteByte(CM730::P_LED_PANNEL,
0x01|0x02|0x04, NULL);
        }
        else if(m_cur_mode == SOCCER)
        {

```

```

        cm730.WriteByte(CM730::P_LED_PANNEL, 0x01,
NULL);
    }
    else if(m_cur_mode == MOTION)
    {
        cm730.WriteByte(CM730::P_LED_PANNEL, 0x02,
NULL);
    }
    else if(m_cur_mode == VISION)
    {
        cm730.WriteByte(CM730::P_LED_PANNEL, 0x04,
NULL);
    }
    else if(m_cur_mode == KIRI)
    {
        cm730.WriteByte(CM730::P_LED_PANNEL,
0x01|0x02, NULL);
    }
    else if(m_cur_mode == KANAN)
    {
        cm730.WriteByte(CM730::P_LED_PANNEL,
0x02|0x04, NULL);
    }
}

if(m_old_btn & BTN_START)
{
    if(m_is_started == 0)
    {
        fprintf(stderr, "Start button pressed.. & started is false..
\n");

        if(m_cur_mode == SOCCER)
        {
            MotionManager::GetInstance()->Reinitialize();
            MotionManager::GetInstance()->SetEnable(true);
            m_is_started = 1;

```



```

        Action::GetInstance()-
>m_Joint.SetEnableBody(true, true);

        Action::GetInstance()->Start(15);
        while(Action::GetInstance()->IsRunning() == true)
usleep(8000);

        /*Head::GetInstance()-
>m_Joint.SetEnableHeadOnly(true, true);
        Walking::GetInstance()-
>m_Joint.SetEnableBodyWithoutHead(true, true);

        MotionManager::GetInstance()-
>ResetGyroCalibration();
        while(1)
        {
            if(MotionManager::GetInstance()-
>GetCalibrationStatus() == 1)
            {
                break;
            }
            else if(MotionManager::GetInstance()-
>GetCalibrationStatus() == -1)
            {
                MotionManager::GetInstance()-
>ResetGyroCalibration();
            }
            usleep(8000);
        }*/
    }
    else if(m_cur_mode == MOTION)
    {
        MotionManager::GetInstance()->Reinitialize();
        MotionManager::GetInstance()->SetEnable(true);
        m_is_started = 1;

        // Joint Enable..
        Action::GetInstance()-
>m_Joint.SetEnableBody(true, true);

```

```

        Action::GetInstance()->Start(1);
        while(Action::GetInstance()->IsRunning() == true)
usleep(8000);
    }
    else if(m_cur_mode == VISION)
    {
        MotionManager::GetInstance()->Reinitialize();
        MotionManager::GetInstance()->SetEnable(true);
        m_is_started = 1;

        // Joint Enable...
        Action::GetInstance()-
>m_Joint.SetEnableBody(true, true);

        Action::GetInstance()->Start(1);
        while(Action::GetInstance()->IsRunning() == true)
usleep(8000);
    }
    else if(m_cur_mode == KIRI)
    {
        MotionManager::GetInstance()->Reinitialize();
        MotionManager::GetInstance()->SetEnable(true);
        m_is_started = 1;

        // Joint Enable...
        Action::GetInstance()-
>m_Joint.SetEnableBody(true, true);

        Action::GetInstance()->Start(1);
        while(Action::GetInstance()->IsRunning() == true)
usleep(8000);
    }
    else if(m_cur_mode == KANAN)
    {
        MotionManager::GetInstance()->Reinitialize();
        MotionManager::GetInstance()->SetEnable(true);
        m_is_started = 1;

```

```

        // Joint Enable...
        Action::GetInstance()-
>m_Joint.SetEnableBody(true, true);

        Action::GetInstance()->Start(1);
        while(Action::GetInstance()->IsRunning() == true)
usleep(8000);
    }
    }
    else
    {
        fprintf(stderr, "Start button pressed.. & started is true..
\n");
    }
}

```

## LAMPIRAN D

### Program Interface

```
/*
 * main.cpp
 *
 * Created on: 2011. 1. 4.
 * Author: robotis
 */

#include <iostream>
#include <stdio.h>
#include <unistd.h>
#include <limits.h>
#include <string.h>
#include <libgen.h>
#include <signal.h>
#include <termios.h>
#include <fcntl.h>
#include <errno.h>

#include "Sensors.h"

#include "mjpg_streamer.h"
#include "LinuxDARwIn.h"

#include "StatusCheck.h"
#include "VisionMode.h"

#ifdef MX28_1024
#define MOTION_FILE_PATH    "../../Data/motion_1024.bin"
#else
#define MOTION_FILE_PATH    "../../Data/motion_4096.bin"
#endif

#define INI_FILE_PATH        "../../Data/config.ini"
#define SCRIPT_FILE_PATH    "script.asc"

#define U2D_DEV_NAME0        "/dev/ttyUSB0"
#define U2D_DEV_NAME1        "/dev/ttyUSB1"
```

```

using namespace std;

bool runallthread = true;
int tari = 0;
int motion_state = 0;
int tari_kanan = 0;
int motion_state_kanan = 0;

LinuxCM730 linux_cm730(U2D_DEV_NAME0);
CM730 cm730(&linux_cm730);

void change_current_dir()
{
    char exepath[1024] = {0};
    if(readlink("/proc/self/exe", exepath, sizeof(exepath)) != -1)
    {
        if(chdir(dirname(exepath)))
            fprintf(stderr, "chdir error!! \n");
    }
}

void sighandler(int sig)
{
    exit(0);
}

int main(void)
{
    signal(SIGABRT, &sighandler);
    signal(SIGTERM, &sighandler);
    signal(SIGQUIT, &sighandler);
    signal(SIGINT, &sighandler);

    //SerialRead();

    change_current_dir();

    minIni* ini = new minIni(INI_FILE_PATH);

```

```

httpd::ini = ini;

//////////////// Framework Initialize //////////////////
if (MotionManager::GetInstance()->Initialize(&cm730) == false)
{
    linux_cm730.SetPortName(U2D_DEV_NAME1);
    if (MotionManager::GetInstance()->Initialize(&cm730) ==
false)
    {
        printf("Fail to initialize Motion Manager!\n");
        return 0;
    }
}

Walking::GetInstance()->LoadINISettings(ini);

MotionManager::GetInstance()->AddModule((MotionModule*)Ac
tion::GetInstance());
MotionManager::GetInstance()->AddModule((MotionModule*)He
ad::GetInstance());
MotionManager::GetInstance()->AddModule((MotionModule*)Wa
lking::GetInstance());

LinuxMotionTimer *motion_timer = new
LinuxMotionTimer(MotionManager::GetInstance());
motion_timer->Start();
////////////////////////////////////

MotionManager::GetInstance()->LoadINISettings(ini);

int firm_ver = 0;
if (cm730.ReadByte(JointData::ID_R_HIP_PITCH,
MX28::P_VERSION, &firm_ver, 0) != CM730::SUCCESS)
{
    fprintf(stderr, "Can't read firmware version from Dynamixel
ID %d!! \n\n", JointData::ID_R_HIP_PITCH);
    exit(0);
}

```

```

        if (0 < firm_ver && firm_ver < 27)
        {
#ifdef MX28_1024
            Action::GetInstance()->LoadFile(MOTION_FILE_PATH);
#else
            fprintf(stderr, "MX-28's firmware is not support 4096
resolution!! \n");
            fprintf(stderr, "Upgrade MX-28's firmware to version
27(0x1B) or higher.\n\n");
            exit(0);
#endif
        }
        else if (27 <= firm_ver)
        {
#ifdef MX28_1024
            fprintf(stderr, "MX-28's firmware is not support 1024
resolution!! \n");
            fprintf(stderr, "Remove '#define MX28_1024' from 'MX28.h'
file and rebuild.\n\n");
            exit(0);
#else

            Action::GetInstance()->LoadFile((char*)MOTION_FILE_PATH);
#endif
        }
        else
            exit(0);

        Action::GetInstance()->m_Joint.SetEnableBody(true, true);
        MotionManager::GetInstance()->SetEnable(true);

        cm730.WriteByte(CM730::P_LED_PANNEL, 0x01 | 0x02 | 0x04,
NULL);

        Action::GetInstance()->Start(15);
        while (Action::GetInstance()->IsRunning()) usleep(8 * 1000);

        Sensors::Serial * serial = new Sensors::Serial(&runallthread);

```

```

serial->start();

char serial_data;
// char serial;
bool panic = false;
bool status = false; //, prev = false;
bool move = true;
while (1)
{
    int out;

    if (cm730.ReadByte(CM730::ID_CM,
CM730::P_ADC2_L, &out, NULL) == CM730::SUCCESS) {
        cout << "out: " << out << endl;
    }
    else {
        cout << "FAIL" << endl;
    }

    serial_data = serial->get_serial();
    cout << "serial : " << serial_data << endl;
    usleep(100);

    int jatuh = 0;
    StatusCheck::Check(cm730, jatuh);

    if (out >=200 || serial_data == 'B' || panic) //data serial dari
Arduiono ketika ON
    {
        printf("GO\n");
        status = true;
    }
    else if (out <=100 && serial_data == 'A' && !panic) //ketika
OFF
    {
        printf("STOP\n");
        status = false;
    }
}

```



```

if (StatusCheck::m_cur_mode == READY)
{
    move = true;
}

if (StatusCheck::m_is_started == 0)
    continue;

switch (StatusCheck::m_cur_mode)
{
case READY:
    break;
case SOCCER:
    motion_state = 0;
    tari = 0;
    tari_kanan = 0;
    motion_state_kanan = 0;
    status = false;
    panic = false;

case MOTION:
    //status = true;
    if (LinuxActionScript::m_is_running == 0) {
Action::GetInstance()->m_Joint.SetEnableBody(true, true);
        if (jatuh) {
            cout << "jatuh" << endl;
            move = true;
            break;
        }
        if (status) {
            printf("GO\n");
            if (move) {
                move = false;
            }
            else {
                if (tari == 0) {
                    printf("Sembah Pembuka A1\n");
                    if (motion_state == 0) {

```

```

motion_state = 1;

Action::GetInstance()->Start(78);
    while
(Action::GetInstance()->IsRunning()) usleep(8 * 10000);
    }
    else if (motion_state == 1) {
        motion_state = 2;

if (tari == 2) {
printf("Zona A2\n");
if (motion_state == 0) {
motion_state = 1;
Action::GetInstance()->Start(92);
while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
else if (motion_state == 1) {
tari = 3;
motion_state = 0;
Action::GetInstance()->Start(93);
while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
}
else if (tari == 3) {
printf("Jalan A3\n");
if (motion_state == 0) {
motion_state = 1;
Action::GetInstance()->Start(96);
while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
else if (motion_state == 1) {
motion_state = 2;
Action::GetInstance()->Start(97);
while (Action::GetInstance()->IsRunning()) usleep(8 * 10000);
}
else if (motion_state == 2) {
motion_state = 3;

```

```

break;
case KIRI:
    status = true;
    panic = true;
    StatusCheck::m_cur_mode = MOTION;
    break;

case KANAN:
    status = true;
    panic = true;
    StatusCheck::m_cur_mode = VISION;
    break;
}

}
}

```

.....*Halaman ini sengaja dikosongkan*.....

## BIODATA PENULIS



Muchammad Ainur Fahd lahir di Bojonegoro pada tanggal 7 September 1995. Sebagai anak terakhir dari dua bersaudara, penulis mengawali kegiatan pendidikan formal di SD Negeri Kauman 1 Bojonegoro, yang kemudian dilanjutkan di SMP Negeri 1 Bojonegoro, SMA Negeri 1 Bojonegoro dan pada tahun 2014 penulis diterima sebagai mahasiswa di Departemen Teknik Elektro ITS dengan Beasiswa Exxon Mobil Cepu Limited. Selama menjalani perkuliahan, penulis aktif dalam

kepanitiaan, kegiatan sosial, organisasi, pelatihan, dan kompetisi. Selain itu, penulis juga aktif dalam Tim Robot VIROSE ITS sejak tahun 2015 sampai dengan tahun 2018. Sejalan dengan semua itu, penulis juga aktif sebagai asisten Laboratorium Elektronika B202. Penulis sangat tertarik dengan kegiatan riset di bidang robotika.

Email : fahad.ee.its@gmail.com  
Hp/WA : 0822 4494 8877  
Facebook : M Ainur Fahad  
LinkedIn : Muchammad Ainur Fahd